ICNLSP 2024

Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP-2024)

19–20 October, 2024

©2024 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
`acl@aclweb.org`

ISBN 979-8-89176-165-0

https://www.icnlsp.org/

Introduction

Welcome to the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)!

ICNLSP is an excellent opportunity for researchers to discuss trends in the field of Natural Language Processing, and to publish their results in the field.

Many topics were discussed through the interesting works presented during the two days of the conference: speech recognition, machine translation, text summarization, sentiment analysis, large language models, natural language understanding, language resources, and other topics etc.

The program committee accepted 47 good papers (long and short ones) which is around 45% of the received submissions, thanks to the high-quality level of the reviews.

"An Introduction to Large Language Models" is the title of a keynote presented by Prof. Gérard Chollet who was chairing the first session: large language models, followed by 06 oral sessions, namely: information extraction and summarization, human-machine interaction and conversational AI, advances in native language identification and text classification, audio, ASR, and TTS, speech emotion recognition and speaker verification and diarization, and data representation.
Prof. Gérard Chollet has clarified many mysterious points on LLMs and discussed their weaknesses, including hallucinations, the black-box nature of their decision-making, and concerns surrounding security and privacy. He presented symbolic AI and knowledge graphs (KGs) as potential solutions.

We thank all participants for their presentations and discussions during the conference.We would like to thank Gérard Chollet and Hugues Sansen (Institut Polytechnique de Paris) for preparing for the special LLM session.

We appreciate the contribution of Dr. Daniel Braun (University of Twente), Dr. Koichi Takeuchi (Okayama University), Dr. Mohammed Mediani (United Arab Emirates University), Prof. Hend Al-Khalifa (King Saud University), Prof. Yücel Saygın (Sabancı Universitesi) and Prof. Nicolas Ballier (Université Paris Cité) for chairing the conference sessions.

Finally, we are grateful to the program committee members for their efforts and commitments.

Mourad Abbas and Abed Alhakim Freihat

Organizers:

General Chair: Dr. Mourad Abbas

Chair: Dr. Abed Alhakim Freihat

Program Chair: Dr. Mourad Abbas

Program Committee:

Ahmed Abdelali, SDAIA, KSA.
Muhammad Abdul-Mageed, University of British Columbia, Canada.
Hend Al-Khalifa, King Saud University, KSA.
Muhammad Al-Qurishi, Elm, KSA.
Mehmet Fatih Amasyalı, Yildiz Technical University, Turkey.
Yuan An, Drexel University, USA.
Christine Balague, Institut Mines-Télécom Business School, France.
Nicolas Ballier, University Paris Cité, France.
Fayssal Bouarourou, University of Strasbourg, France.
Pierrette Bouillon, University of Geneva, Switzerland.
Daniel Braun, University of Twente, Netherlands.
Hadda Cherroun, Amar Telidji University, Algeria.
Gérard Chollet, CNRS, France.
Najim Dehak, Johns Hopkins University, USA.
Abed Alhakim Freihat, University of Trento, Italy.
Mounim El Yacoubi, Telecom SudParis, France.
Ahmed Guessoum, USTHB, Algeria.
Kais Haddar, Faculty of Sciences of Sfax, Tunisia.
Fouzi Harrag, Ferhat Abbas University, Algeria.
Kristiina Jokinen, University of Helsinki, Finland.
Ken Kawamura, Revelata Inc., USA.
Eric Laporte, Gustave Eiffel University, France.
Mohammed Mediani, UAE University, UAE.
Abdelouahab Moussaoui, Ferhat Abbas University, Algeria.
Hermann Ney, RWTH Aachen University, Germany.
Robert Nickel, Bucknell University, USA.
Ahmed Rafea, American University in Cairo, Egypt.
Hugues Sansen, Institut Polytechnique de Paris, France.
Yücel Saygin, Sabanci University, Turkey.
Thomas Schmidt, University of Regensburg, Germany.
Nasredine Semmar, CEA, France.
Peter Sullivan, University of British Columbia, Canada.
Irina Temnikova, Big Data for Smart Society Institute, Bulgaria.
María Inés Torres, University of the Basque Country, Spain.

Jan Trmal, AppTek, USA.
Nikos Tsourakis, University of Geneva, Switzerland.
Guanqun Yang, Stevens Institute of Technology, USA.
Fayçal Ykhlef, CDTA, Algeria.
Wajdi Zaghouani, HBKU, Qatar.
Zbyněk Zajíc, University of West Bohemia, Czechia.

Invited Speakers:

Prof. Gérard Chollet, CNRS, France.

Invited Talk

Prof. Gérard Chollet was granted a PhD in Computer Science and Linguistics from the University of California, Santa Barbara. He taught at Memphis State University and University of Florida before joining CNRS. In 1981, he took in charge the speech research group of Alcatel. In 1983, he joined a CNRS research unit at ENST (Institut Mines-Telecom). In 1992, he participated to the development of IDIAP, a research laboratory of the 'Fondation Dalle Molle' in Martigny, Switzerland. From 1996 to 2012, he was back full time at ENST. He supervised more than forty doctoral theses. CNRS decided in 2012 to grant him an emeritus status within SAMOVAR (Télécom-SudParis). His main research interests are in phonetics, automatic audio-visual speech processing, spoken dialog systems, multimedia, pattern recognition, biometrics, privacy-preserving digital signal processing, speech pathology and speech training aids. His main publications are available from his Google Scholar Citations profile.

An Introduction to Large Language Models

Prof. Gérard Chollet, CNRS, France.

In the tutorial "An Introduction to Large Language Models," key weaknesses of LLMs were discussed, including hallucinations, the black-box nature of their decision-making, indecisiveness, and concerns surrounding security and privacy. To address these challenges, symbolic AI and knowledge graphs (KGs) were introduced as potential solutions. The integration of LLMs with KGs can enhance transparency and reliability in AI outputs, leveraging structured knowledge to mitigate errors and improve decision-making. A significant focus was placed on an embedded solution developed within the e-ViTA project (https://www.e-vita.coach/), specifically designed to address privacy issues. This implementation showcases how KGs can provide a framework for safeguarding sensitive information while still allowing LLMs to function effectively. By utilizing KGs, users can access more interpretable and contextually accurate information without compromising data security. The tutorial emphasized the importance of unifying LLMs and KGs to create more robust AI systems that are not only powerful but also trustworthy and secure. Through this approach, the future of AI can balance innovation with ethical considerations, paving the way for more responsible applications in various domains.

Table of Contents

Leveraging Annotator Disagreement for Text Classification 1
Jin Xu, Mariët Theune and Daniel Braun

Native language Identification for Arabic Language Learners using Pre-trained Language Models 11
Mohamed Amine Cheragui, Mourad Abbas and Mohammed Mediani

Generative Adversarial Network based Neural Vocoder for Myanmar End-to-End Speech Synthesis 19
Aye Mya Hlaing and Win Pa Pa

Detecting ChatGPT-Generated Text with GZIP-KNN: A No-Training, Low-Resource Approach . 25
Matthias Berchtold, Sandra Mitrovic, Davide Andreoletti, Daniele Puccinelli and Omran Ayoub

Data Bias According to Bipol: Men are Naturally Right and It is the Role of Women to Follow Their Lead 34
Irene Pagliai, Goya van Boven, Tosin Adewumi, Lama Alkhaled, Namrata Gurung, Isabella Södergren and Elisa H Barney Smith

Improved Spoken Emotion Recognition With Combined Segment-Based Processing And Triplet Loss 47
Dejan Porjazovski, Tamas Grosz and Mikko Kurimo

EEG Signal Analysis for Multimodal Simple Concepts Decoding 55
Sergio Guillén Jiménez, Lorenzo J. Tardón, Ana M Barbancho and Isabel Barbancho

Modeling Score Estimation for Japanese Essays with Generative Pre-trained Transformers 64
Boago Okgetheng and Koichi Takeuchi

CliqueCorex: A Self-supervised Clique-based Anchored Topic Model 74
Sami Diaf

Double Decoder: Improving latency for Streaming End-to-end ASR Models 83
Riqiang Wang, Shreekantha Nadig, Daniil Kulko, Simon Vandieken, Chia-tien Chang, Seyyed Saeed Sarfjoo and Jonas Robertson

Personalised Abusive Language Detection Using LLMs and Retrieval-Augmented Generation . . 92
Tsungcheng Yao, Ernest Foo and Sebastian Binnewies

Large-scale Summarization of Chat Transcripts in the Absence of Annotated Summaries 99
Pratik K. Biswas

Real-Time Speech-Driven Avatar Animation by Predicting Facial landmarks and Deformation Blendshapes 109
Juan Camilo Vasquez-Correa, Santiago Moreno-Acevedo, Ander Gonzalez-Docasal, Aritz Lasarguren, Jone Lòpez, Egoitz Rodriguez and Aitor Álvarez

Speech Emotion Recognition for Call Centers using Self-supervised Models: A Complete Pipeline for Industrial Applications 119
Juan M. Martín-Doñas, Asier López Zorrilla, Mikel deVelasco, Juan Camilo Vasquez-Correa, Aitor Álvarez, Maria Inés Torres, Paz Delgado, Ane Lazpiur, Blanca Romero and Irati Alkorta

Probing Whisper Predictions for French, English and Persian Transcriptions 129
Nicolas Ballier, Léa Burin, Behnoosh Namdarzadeh, Sara B Ng, Richard Wright and Jean-Baptiste Yunès

Sawaal: A Framework for Automatic Question Generation in Urdu 139

Maria Rahim and Shakeel Ahmed Khoja

Thonburian Whisper: Robust Fine-tuned and Distilled Whisper for Thai 149
Zaw Htet Aung, Thanachot Thavornmongkol, Atirut Boribalburephan, Vittavas Tangsriworakan, Knot Pipatsrisawat and Titipat Achakulvisut

Dual-Task Learning for AI-Generated Medical Text Detection and Named Entity Recognition . . 157
Saja Al-Dabet, Ban Alomar, Sherzod Turaev and Abdelkader Nasreddine Belkacem

Comparative Analysis of Modality Fusion Approaches for Audio-Visual Person Identification and Verification 168
Aref Farhadipour, Masoumeh Chapariniya, Teodora Vukovic and Volker Dellwo

PoliTun: Tunisian Political Dataset for Detecting Public Opinions and Categories Orientation . . . 178
Chayma Fourati, Roua Hammami, Chiraz Latiri and Hatem Haddad

Efficient Few-shot Learning for Multi-label Classification of Scientific Documents with Many Classes 186
Tim Schopf, Alexander Blatzheim, Nektarios Machner and Florian Matthes

A Comparison of Different Tokenization Methods for the Georgian Language 199
Beso Mikaberidze, Temo Saghinadze, Guram Mikaberidze, Raphael Kalandadze, Konstantine Pkhakadze, Josef van Genabith, Simon Ostermann, Lonneke van der Plas and Philipp Müller

GemST: Continual Learning for End-to-End Speech-to-Text Translation 209
Pranav Karande, Balaram Sarkar and Chandresh K Maurya

CASCA: Leveraging Role-Based Lexical Cues for Robust Multimodal Speaker Diarization via Large Language Models 214
William K Nehrboss

Deep Information Maximisation to Mitigate Information Loss in Text Independent Speaker Verification 225
Nipun Thejan Fonseka and Nirmal Sankalana

Context-Aware Question Answering in Urdu 233
Samreen Kazi and Shakeel Ahmed Khoja

Human and Machine: Language Processing in Translation Tasks 243
Hening Wang, Leixin Zhang and Ondřej Bojar

Asking the Right Questions: Exploiting Hidden Interactions in a Generative Framework for Multilingual, Multitask Classification 251
Sebastian-Antonio Toma, Camelia Lemnaru, Vlad Andrei Negru and Rodica Potolea

Resolving Gender Biases in LLMs at Inference Time with Novel Dijkstra's-based K-Explorers Neural Network Traversal (KeNNT) 267
Hanav Modasiya

Semantically Enriched Text Generation for QA through Dense Paraphrasing 279
Timothy Obiso, Bingyang Ye, Kyeongmin Rim and James Pustejovsky

Introducing wwm-german-18k - Can LLMs Crack the Million? (Or Win at Least 500 Euros?) . . 287
Matthias Aßenmacher, Luis Karrlein, Philipp Schiele and Christian Heumann

Native Language Identification Improves Authorship Attribution 297
Ahmet Yavuz Uluslu, Gerold Schneider and Can Yildizli

Design and Comparison of Arabic Negotiation Bots Using LLMs versus Seq2Seq Models with Reinforcement Learning 304
Ahmad Hajj, Yasmine A Abu Adla, Samah Albast, Hazem Hajj, Shady Elbassuoni, Wassim El Hajj and Khaled Shaban

Enhancing LLM-based Arabic Negotiation by Fine Tuning on Dialogue Shortcomings 319
Yasmine A Abu Adla, Hazem Hajj, Shady Elbassuoni, Khaled Shaban and Wassim El Hajj

The Qiyas Benchmark: Measuring ChatGPT Mathematical and Language Understanding in Arabic 343
Shahad Al-Khalifa and Hend Al-Khalifa

Analyzing Politeness in Arabic Tweets: A Preliminary Study 352
Hend Al-Khalifa, Nadia Ghezaiel and Maria Bounnit

FeruzaSpeech: A 60 Hour Uzbek Read Speech Corpus with Punctuation, Casing, and Context . . 360
Anna Povey and Katherine Povey

Bulgarian Grammar Error Correction with Data Augmentation and Machine Translation Techniques 365
Bozhidar Klouchek and Riza Batista-Navarro

On Barriers to Archival Audio Processing 377
Peter R Sullivan and Muhammad Abdul-Mageed

Conversational Exploratory Search of Scholarly Publications Using Knowledge Graphs 384
Phillip Schneider and Florian Matthes

A Hybrid Retrieval Approach for Advancing Retrieval-Augmented Generation Systems 397
Nguyen Nam Doan, Aki Härmä, Remzi Celebi and Valeria Gottardo

Investigating Gender Bias in Large Language Models Through Text Generation 410
Shweta Soundararajan and Sarah Jane Delany

Improving Long-term F0 representation using post-processing techniques 425
Crisron Rudolf Lucas, Diptasree Debnath and Andrew Hines

Text-to-Speech in Azerbaijani Language via Transfer Learning in a Low Resource Environment . 434
Dzhavidan Zeinalov, Bugra Sen and Firangiz Aslanova

SG-RAG: Multi-Hop Question Answering With Large Language Models Through Knowledge Graphs 439
Ahmmad O. M. Saleh, Gokhan Tur and Yucel Saygin

Linking Quran and Hadith Topics in an Ontology using Word Embeddings and Cellfie Plugin . . . 449
Ibtisam Khalaf Alshammari, Eric Atwell and Mohammad Ammar Alsalka

Medical Information Extraction with Large Language Models 456
Raffaello Fornasiere, Nicolò Brunello, Vincenzo Scotti and Mark Carman

Leveraging Annotator Disagreement for Text Classification

Jin Xu
University of Twente
jinxu130817@gmail.com

Mariët Theune
University of Twente
Human-Media Interaction
m.theune@utwente.nl

Daniel Braun
University of Twente
Industrial Engineering and
Business Information Systems
d.braun@utwente.nl

Abstract

It is common practice in text classification to only use one majority label for model training even if a dataset has been annotated by multiple annotators. Doing so can remove valuable nuances and diverse perspectives inherent in the annotators' assessments. This paper proposes and compares three different strategies to leverage annotator disagreement for text classification: a probability-based multi-label method, an ensemble system, and instruction tuning. All three approaches are evaluated on the tasks of hate speech and abusive conversation detection, which inherently entail a high degree of subjectivity. Moreover, to evaluate the effectiveness of embracing annotation disagreements for model training, we conduct an online survey that compares the performance of the multi-label model against a baseline model, which is trained with the majority label. The results show that in hate speech detection, the multi-label method outperforms the other two approaches, while in abusive conversation detection, instruction tuning achieves the best performance. The results of the survey also show that the outputs from the multi-label models are considered a better representation of the texts than the single-label model.

1 Introduction

Employing multiple annotators for data annotation and afterwards using the majority annotation for model training is a widely adopted practice to mitigate biases and allow for error detection and correction (Sabou et al., 2014).

However, such procedures also remove genuine disagreement between annotators that can provide valuable insights, e.g. for subjective tasks like detection of hate speech, emotions, or sexism, but also for more objective tasks like legal or medical decision making. In recent years, the practice of only considering majority annotations has been increasingly criticized and many researchers have started to advocate for better ways to deal with disagreement between annotators (Basile et al., 2021; Uma et al., 2021; Plank, 2022; Braun, 2024).

In this article, we propose three different strategies to leverage annotator disagreement during the training of text classification models: a probability-based multi-label approach, an ensemble system approach, and an instruction tuning approach. We compare these strategies against a baseline model that is trained on the majority labels derived from the multiple annotations. We choose two text classification tasks which inherently entail a high degree of subjectivity for the evaluation: hate speech detection and abuse detection in conversations. In our chosen datasets (Toraman et al., 2022; Cercas Curry et al., 2021), these two tasks exhibit different complexity and difficulty in terms of the label space: while the hate speech detection dataset contains binary labels, the abusive conversation detection dataset is not only annotated with abusive / non-abusive but also the severity of the abuse.

Our first approach tackles the tasks as a probability-based multi-label text classification problem. Instead of predicting specific labels to one instance, the model provides a probability distribution. The second approach imitates the process of annotation from multiple annotators with an ensemble system. The ensemble system consists of many sub-models, each of which is trained on different labels to capture the diverse viewpoints embedded in the annotations. Thirdly, we use instruction tuning. Specifically, we use a pre-trained generative model and inject explicit guidance into the training process to customize the model's behavior. The performance of the proposed models is compared using cross entropy. To evaluate the effectiveness of incorporating multiple labels, we also conduct an online survey. This survey aims to investigate human preferences between the outputs generated by the multi-label model and a baseline model.

The results show that on the hate speech dataset, the multi-label method outperforms the ensemble system and instruction tuning. Conversely, instruction tuning is the best-performing method on the abusive conversation dataset. Through multinomial test, the outputs from the multi-label model are considered more reasonable than those from the baseline model to characterize samples from the online survey. This proves the effectiveness of leveraging annotation disagreements for model training.

2 Related Work

2.1 Sources of Disagreement

Disagreement in annotations can originate from different sources. Natural language can be inherently complex and interpreted in multiple ways within a given context (Poesio, 2020). There are many subjective elements which may add an additional layer of intricacy to the understanding of texts, such as sentiments, opinions or nuanced expressions. Therefore, it is common that there are divergent interpretations among annotators. Furthermore, some sentences and even the definition of labels may contain vague or ambiguous statements (Russell et al., 2008), making it challenging for annotators to reach an agreement.

However, annotators themselves and their background can also have significant impact on the annotation results (Davani et al., 2022). Through post-annotation interviews, Patton et al. (2019), for example, showed that annotators who come from communities discussed in gang-related tweets are more likely to rely on their lived experiences in the process of annotating when compared to graduate student researchers. This divergence results in distinct label judgments. Luo et al. (2020) found that the political affiliation of annotators can significantly shape how they assess and annotate political stances.

2.2 Handling Disagreement

Majority voting involves aggregating annotations by selecting the label that the majority of annotators agree upon. Majority voting is easy to understand and implement and tends to perform well when the annotators share unanimous perspectives (Uma et al., 2021). However, the employment of a majority voting method in annotation processes can unintentionally obscure nuanced viewpoints, especially for groups that are underrepresented in annotator pools (Prabhakaran et al., 2021). To address this concern, it is important to ensure a diverse representation among annotators to foster a more comprehensive understanding of various perspectives, particularly those from underrepresented demographics (Wan et al., 2023).

Some studies have introduced alternative methods to majority voting in order to incorporate annotator disagreement in model training. Chou and Lee (2019) modelled the label uncertainty and annotator idiosyncrasy simultaneously by using both hard label (majority voting) and soft label (the distribution of annotations). The results showed that the soft label contains useful information that significantly boosts the model performance. Fornaciari et al. (2021) proposed a multi-task neural network that was trained on soft label distribution over annotator labels. By integrating a divergence measurement between soft label and “true” label vector into the loss functions, they effectively mitigated overfitting and therefore improved model performance. Davani et al. (2022) introduced multi-annotator models where each annotator’s judgements were regarded as independent sub-task with a shared common representation of the annotation task. This approach enables to preserve and model the internal consistency in each annotator’s label. It also incorporates the systematic disagreements with other annotators. Similarly, the network architecture introduced by Guan et al. (2018) individually models annotation experts. In this approach, each expert’s model weight is calculated independently, and these individual weights are then averaged to facilitate ensemble recognition. To include the knowledge from annotators, Fayek et al. (2016) employed neural networks to build an ensemble system that consists of many models, with each model representing one annotator. Then the final results are obtained by combining the individual model outputs.

Although the approaches outlined above have improved the performance by leveraging annotation disagreements, they remained limited to identifying the majority label. The outputs, in the form of “soft labels” (probability distribution over labels), were still aggregated to single labels as final predictions. There is limited research focusing on evaluating the effectiveness of embracing multiple labels.

3 Datasets

In this section, the two datasets that have been used in this study will be briefly introduced.

3.1 Hate Speech

The first dataset is the "Large-Scale Hate Speech Dataset"[1] published by Toraman et al. (2022). It consists of a total of 100,000 tweets (7,000 training, 1,500 validation, and 1,500 testing). Each tweet in the dataset is annotated by five annotators that have been selected randomly from a panel of 20 annotators. According to the annotation guidelines utilized by Sharma et al. (2018), tweets are categorized as "Hate" if they target, incite violence against, threaten, or advocate for physical harm towards an individual or a group of people based on identifiable trait or characteristic. If tweets humiliate, taunt, discriminate against, or insult an individual or a group of people, they are annotated as "Offensive". In the absence of these criteria, the tweets are labeled as "Normal" .

3.2 Abuse Conversation

The second dataset is the "Abuse in Conversational AI" dataset[2] (hereinafter referred to as "abusive conversation dataset") published by Cercas Curry et al. (2021). The data was collected from conversations between users and conversational AI systems, and consists of 2501 samples as training data, 831 as validation data and 853 as testing data. The data was annotated using an unbalanced rating scale proposed by Poletto et al. (2019), in which inputs are labelled on a scale from Not abusive, Ambiguous, Mildly abusive, Strongly abusive to Very strongly abusive. This annotation scheme offers insights into not only the presence of abusive content, but also the severity of the abuse. In the annotation process, eight annotators were recruited, and each example is annotated by a minimum of three annotators.

4 Methodology

4.1 Baseline model

The baseline model for this study is trained on the "ground truth" label that is aggregated via majority voting. Given BERT's (Devlin et al., 2019) notable performance in contextual understanding, we choose it as the pre-trained model. Since the baseline model outputs a single label, we augment its architecture by adding a fully connected layer to the last hidden state, thereby adapting the model structure to the specific prediction task.

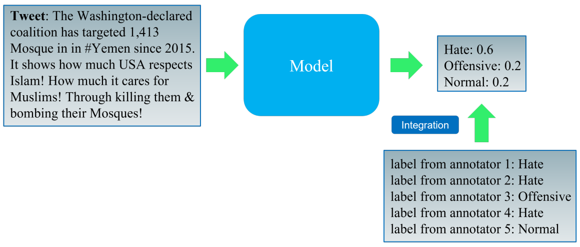

Tweet: The Washington-declared coalition has targeted 1,413 Mosque in in #Yemen since 2015. It shows how much USA respects Islam! How much it cares for Muslims! Through killing them & bombing their Mosques!
Model
Hate: 0.6
Offensive: 0.2
Normal: 0.2
Integration
label from annotator 1: Hate
label from annotator 2: Hate
label from annotator 3: Offensive
label from annotator 4: Hate
label from annotator 5: Normal

Figure 1: The framework of model training within the probability-based multi-label method.

4.2 Probability-based multi-label method

The task of identifying hate speech or abusive conversation can be regarded as a multi-label text classification problem, where a given piece of text can be associated with one or multiple labels simultaneously. Unlike the traditional approaches that assign one or several exclusive labels to the input text (Jiang and Nachum, 2020), our model predicts the probability of each label being associated with the given text. The approach is illustrated in Figure 1. The model is trained on the probability distribution across different labels which is derived from individual annotations. Like the baseline model, the multi-label model also is based on BERT but fine-tuned with different types of target labels.

4.3 Ensemble system

In the annotation process, multiple labels are assigned by different annotators. Inspired by this process, we propose an ensemble system consisting of several sub-models. As shown in Figure 2, each sub-model is based on a BERT model that is fine-tuned individually on its respective set of labels. For each sub-model, the input is the text from one sample and the output is a multidimensional vector where each dimension corresponds to one category. After that, this vector is transformed by the SoftMax function and the dimension with highest probability is identified as the output of the sub-model. Finally, the predictions from all sub-models are combined and converted into a probability distribution of three- or five-dimensional vector.

In the abusive conversation dataset, the annotators assigned for each sample are clearly specified

---

[1] https://github.com/avaapm/hatespeech/tree/master/dataset_v1

[2] https://github.com/amandacurry/convabuse/tree/main

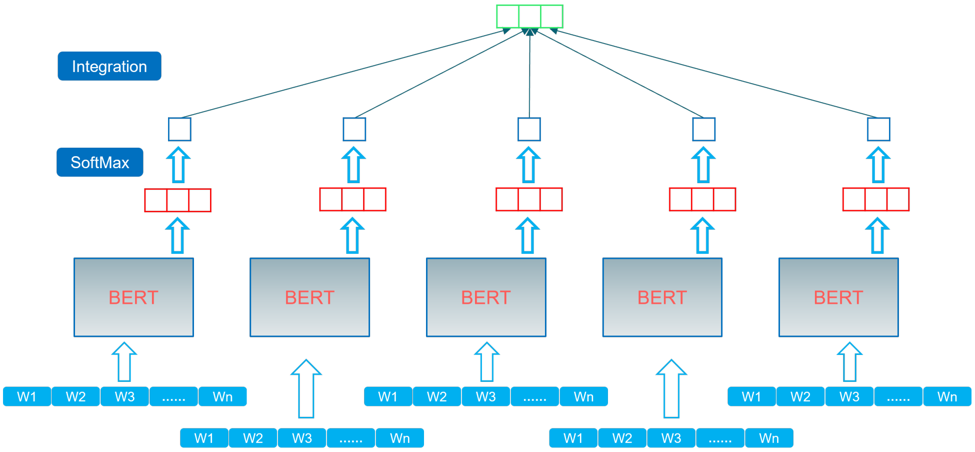
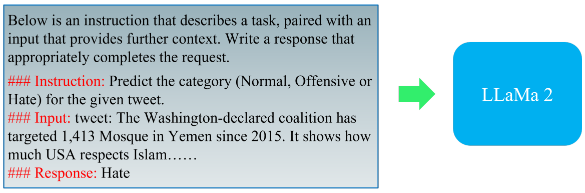
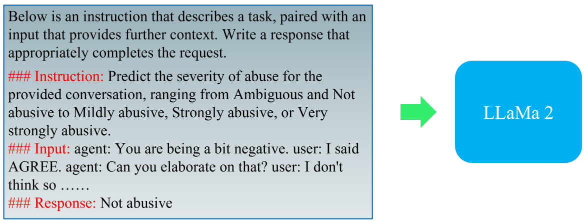


Figure 2: Fine-tuning BERT individually as sub-models within the ensemble system.



```
Below is an instruction that describes a task, paired with an
input that provides further context. Write a response that
appropriately completes the request.
### Instruction: Predict the category (Normal, Offensive or
Hate) for the given tweet.
### Input: tweet: The Washington-declared coalition has
targeted 1,413 Mosque in Yemen since 2015. It shows how
much USA respects Islam……
### Response: Hate
```

Figure 3: Fine-tuning LLaMa 2 as a sub-model with instruction tuning in the hate speech dataset.



```
Below is an instruction that describes a task, paired with an
input that provides further context. Write a response that
appropriately completes the request.
### Instruction: Predict the severity of abuse for the
provided conversation, ranging from Ambiguous and Not
abusive to Mildly abusive, Strongly abusive, or Very
strongly abusive.
### Input: agent: You are being a bit negative. user: I said
AGREE. agent: Can you elaborate on that? user: I don't
think so ……
### Response: Not abusive
```

Figure 4: Fine-tuning LLaMa 2 as a sub-model with instruction tuning in the abusive conversation dataset.

and identifiable. Therefore, within the ensemble system, each sub-model represents one specific annotator and is trained on that annotator's labels. By contrast, in the hate speech dataset, each sample is labeled by five anonymous annotators. Despite the anonymity, training a model with such labels can potentially increase the robustness of sub-models since it helps to reduce the biases or inconsistencies introduced by individual annotators (Frenay and Verleysen, 2014). Furthermore, the resulting labels are likely to reflect a diverse range of perspectives and interpretations of the data. Training sub-models on these diverse annotations can capture the variability in annotator judgments and enhance the model's ability to generalize across different viewpoints (Audhkhasi and Narayanan, 2013). Since the sub-models can show varying performances in the training and validation processes, typically, the top n (n≥3) best-performing sub-models are chosen to determine the final output. The ranking is based on their accuracy on the validation data.

4.4 Instruction tuning

Instruction tuning is the process of fine-tuning LLMs in a supervised fashion on a dataset consisting of pairs of instructions and outputs. The key idea is to provide the model with explicit instructions to enhance its performance and align it with specific objectives. Unlike traditional training approaches where models learn from data alone, instruction tuning injects explicit guidance into the training process. This approach allows for explicit customization of the model's behavior. In this study, we ask the model to predict the class of hate speech or abusive conversation based on the input we construct. The input contains the task description, the instruction, the original text, and the annotation from one specific annotator (i.e. *not* the majority label). The approach of fine-tuning LLaMa 2 via instruction tuning on the two datasets is presented in Figure 3 and Figure 4. On the left sides of the figures are the inputs fed into the pre-trained model. The input comprises the following four components: scenario description, instruction, text input and response.

Like the ensemble system, top n (n≥3) best-performing sub-models are selected to contribute to the final predictions.

	Training	**Validation**	**Testing**
Hate speech	0.7613	0.7569	0.7638
Abusive conversation	0.8861	0.9680	0.9834

Table 1: The average cross entropy of the probability-based multi-label model on two tasks.

4.5 Evaluation

4.5.1 Cross entropy

The final output of our proposed models is a probability distribution across different labels, and in this scenario, a single "ground truth" label is no longer applicable for model evaluation. Instead, we use cross entropy to compare the distribution of annotations with model output. Cross entropy is one kind of statistical distance which measures how a probability distribution is different from a reference probability distribution. In the field of NLP, it has been used to quantify how well the model's predicted distribution matches the annotation distribution (Pavlick and Kwiatkowski, 2019).

4.5.2 Online survey

Using cross entropy to evaluate the effectiveness of training models with multiple labels against models that only rely on the majority label is impossible due to the format disparity between the outputs generated. To bridge this gap, we conduct an online survey where participants specify their preference between annotations generated from the probability-based multi-label model and the baseline model. For each dataset, we select 10 samples, each featuring two annotations. Both annotations are in the form of probability distributions across different labels. One is generated from the baseline model trained with majority labels, which is, however, used to generate a probability distribution in the phase of inference. The other one is from the probability-based multi-label model. This model has the same structure as the baseline model and their only difference is the labels they were trained on. For each sample, participants are required to indicate which annotation they find is more reasonable to characterize the tweet or the abusive conversation.

5 Results

5.1 Multi-Label Method

Table 1 show the performance of the multi-label model on the two datasets. In this approach, the model demonstrates superior performance on the hate speech dataset compared to the abusive conversation dataset. In particular, the cross entropy for the hate speech dataset is 0.7638, while this value for the abusive conversation dataset is 0.9834. With a parameter size of 110 million, the multi-label model benefits from the extensive training data in hate speech dataset to optimize and align itself with the downstream task. In contrast, there are only 2501 training samples in the abusive conversation dataset, which can easily lead to overfitting in the process of training. The multi-label model exhibits relatively consistent losses across training, validation, and testing data in the hate speech dataset, indicating a good fit without signs of underfitting or overfitting. By comparison, in the abusive conversation dataset, losses during validation and testing are noticeably higher than during training. When the model encounters unseen data in validation and testing phases, the loss can be relatively high due to the lack of generalization.

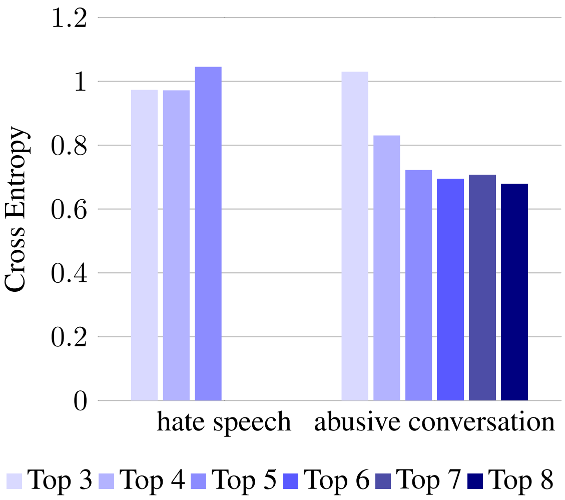


Figure 5: Comparison of the ensemble system's performances on two tasks.

5.2 Ensemble system

Figure 5 shows the ensemble system's performances on the two datasets. In the testing phase, we select the top-performing sub-models based on their validation accuracies. The ensemble system performs better on the abusive conversation dataset than on the hate speech dataset. Specifically, in the hate speech dataset, the best performance is achieved by the top 3 sub-models and the corresponding overall cross entropy loss is 0.9720. Conversely, the best overall cross entropy for the abusive conversation dataset is 0.6782, achieved with

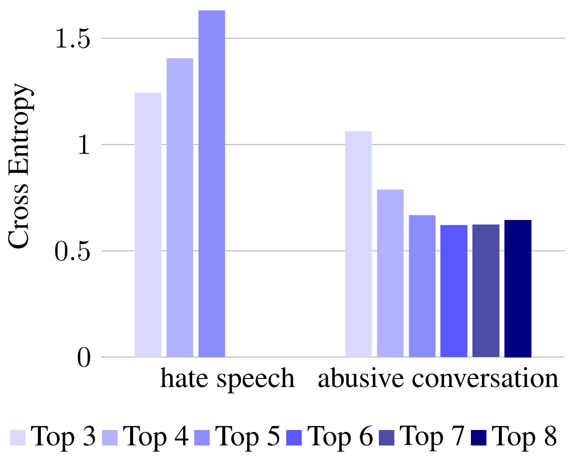


Figure 6: Comparison of instruction tuning's performances on two tasks.

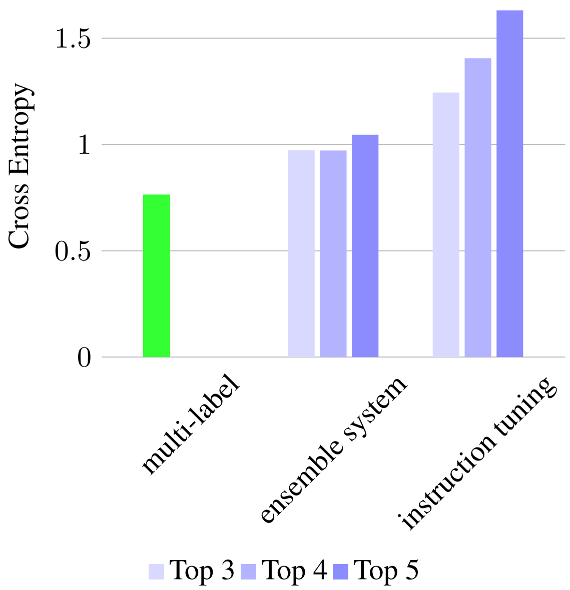


Figure 7: Comparison of different models' performances on the hate speech dataset.

the top 8 (all) sub-models. The ensemble system is designed to simulate the process of annotation and has a large parameter size. Despite being trained on a substantially larger dataset, this method performs less effectively for the hate speech dataset. In this dataset, 20 annotators contribute, with each sample being annotated by five randomly assigned annotators, which means the five annotators for all the samples are not always the same individuals. As a result, one single sub-model may struggle to learn the specific characteristics of each annotator from the data. By contrast, in the abusive conversation dataset, there are eight annotators in total and for each sample it is clearly indicated which annotators are assigned for the annotation task. In this context, each sub-model is designed to emulate an individual annotator. Consequently, the ensemble system integrates the unique insights from each individual annotator, as represented by the sub-models.

5.3 Instruction Tuning

Figure 6 shows the performance of the instruction tuning approach. In this approach, even though with a considerably smaller training data size, the model's performance on the abusive conversation dataset is significantly better compared to the hate speech dataset. In the hate speech dataset, the best performance is achieved by the top 3 sub-models, with a cross entropy of 1.2445. By contrast, the lowest cross entropy in the abusive conversation dataset, achieved by the top 6 sub-models, is 0.6200. Unlike traditional machine learning or deep learning algorithms, one of the most evident advantages of instruction tuning is that it does not require much training data. Even though there are only 2501 training samples in the abusive conversation dataset, it is already sufficient to fine-tune the model and enable it to grasp the specific patterns and knowledge within the data. With this limited dataset, the pre-trained model selectively activates or deactivates certain neurons in the neural network, which serves an important role in revealing or concealing some functions embedded in LLaMa 2. Although the hate speech dataset contains a large amount of training data, the individual samples annotated by specific annotators remain unknown, which presents a challenge for the model in terms of fitting and learning patterns from the data.

5.4 Comparison

As shown in Figure 7, the multi-label method outperforms the other approaches on the hate speech dataset. The reason behind this might be the aforementioned issue in this dataset: the five annotators assigned to each sample are anonymous. Both the ensemble system and instruction tuning were trained using the same paradigm, where sub-models were fine-tuned individually on their respective labels. On the contrary, the multi-label model only relied on the probability distribution across different classes as the target, circumventing the issue with annotator anonymity. Furthermore, the hate speech dataset is big enough to fine-tune the BERT model.

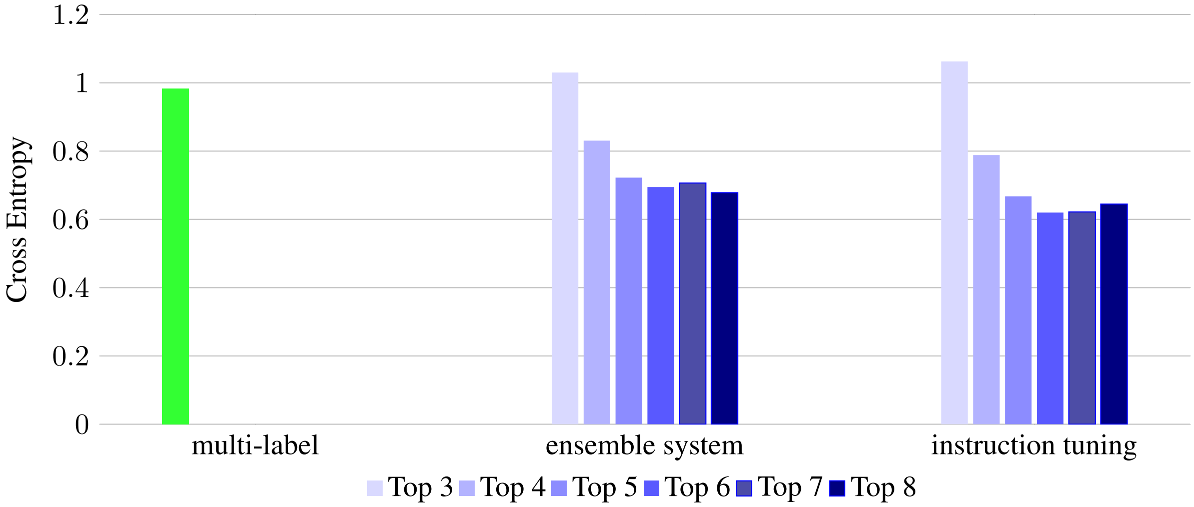


Figure 8: Comparison of different models' performances on abusive conversation dataset.

Figure 8 shows that on the abusive conversation dataset the multi-label method performs worst. The size of this dataset is relatively small, which can result in overfitting during fine-tuning. The ensemble system consists of sub-models, with each tailored to predict annotations from a specific annotator. With multiple sub-models making their own decisions independently and contributing to the final prediction, the ensemble system can mitigate the bias brought by overfitting. Since instruction tuning does not have a high requirement for dataset size, it performs slightly better than the ensemble system.

5.5 Online survey

In exploring the probability distribution preference, we recruited 36 participants for the online survey. The multinomial test (Read and Cressie, 2012) is employed since there are three possible preference options. The details of the results are outlined in Table 2. From this table, the multinomial tests for the multi-label model on two datasets are statistically significant, with the p-value of 0.0000. This means there is a notable disparity among the three categories being compared. Individuals generally favor the multi-label model as a more reasonable representation to characterize tweets or conversations. The results indicate the effectiveness of leveraging annotation disagreements in model training.

6 Conclusion

In this paper, we proposed and compared three approaches to incorporate diverse annotations in the training of ML models: a probability-based multi-label method, an ensemble system, and instruction tuning. All three approaches take the individual labels from all annotators into account for model training in different ways, rather than only depending on an assumed "ground truth" label. In this way, the output includes a rich diversity of perspectives from annotators. We applied the proposed models on two datasets, which correspond to two tasks: hate speech detection and abuse detection in conversational AI. The two datasets show discrepancies in terms of data size, classification difficulty, the number of annotators involved in each sample, and their anonymity levels. Results show that on the hate speech dataset, the multi-label method demonstrates the highest performance among the three models, while instruction tuning achieves the lowest loss on the abusive conversation dataset. Lastly, an online survey was conducted to evaluate the performance of the probability-based multi-label model in comparison to the baseline model. The online survey investigated individuals' preference between the distributions generated from the multi-label model and the baseline model. The evaluation of the survey results showed that the distribution generated from the multi-label model is considered more reasonable to characterize the texts compared to the baseline model. In the future, we would like to explore some methods or techniques to mitigate the class-imbalanced issue in the dataset. For example, there have been many popular algorithms that contribute to a relatively class-balanced dataset by over sampling (Chawla et al., 2002) or down sampling (Wilson, 1972). We would also like to work on investigating automatically generated prompts.

	Hate speech			Abusive conversation		
Preference	**Counts**	**Proportion**	**P-value**	**Counts**	**Proportion**	**P-value**
Baseline	118	0.3278	0.6078	152	0.4222	0.0003
Multi-label model	198	0.5500	**0.0000**	194	0.5389	**0.0000**
No difference	44	0.1222	1.0000	14	0.0389	1.0

Table 2: Multinomial test for probability distribution preference on two datasets.

Recent research has demonstrated that a concrete prompt, which consists of several discreate tokens, may not always be the most effective prompt to instruct the behavior of the model (Liu et al., 2023). Conversely, continuous embeddings of prompts, which might lack immediate human interpretability, make sense for the model itself (Li and Liang, 2021; Subramani et al., 2019).

Limitations

There are some limitations to the experiments. Firstly, the ensemble system showed to be not suitable for the hate speech dataset, where the five annotators assigned to each sample are not fixed. In this dataset, each set of annotations used for training a sub-model can comprise annotations from multiple individuals. As a result, it becomes impossible for the sub-models to capture the specific characteristics of each annotator embedded in the annotations.

Secondly, both datasets in this study suffer from class-imbalanced problem, which can have an adverse impact on model training. When trained on a class-imbalanced dataset, the model primarily focuses on the samples from the majority class and neglect those from the minority class, as that is an efficient strategy for minimizing the training loss. Another limitation is the inconsistency among annotators, which can introduce noise into the dataset and weaken model performance. Since our dataset lacked identifiable annotators, it was not possible to model individual annotator bias or assess inter-annotator agreement comprehensively. This constrains our ability to account for subjective variations in labeling.

Thirdly, we only leverage manually created prompts, which may introduce subjectivity and bias based on the prompt maker's perspective (Tian et al., 2023). It has been proved that manually created prompts suffer from a high degree of instability and a minor change in the prompt can result in substantial discrepancies in the model' s performance (Liu et al., 2023).

References

Kartik Audhkhasi and Shrikanth Narayanan. 2013. A Globally-Variant Locally-Constant Model for Fusion of Labels from Multiple Diverse Experts without Using Reference Labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):769–783.

Valerio Basile, Federico Cabitza, Andrea Campagner, and Michael Fell. 2021. Toward a Perspectivist Turn in Ground Truthing for Predictive Computing. *CoRR*, abs/2109.04270. ArXiv: 2109.04270.

Daniel Braun. 2024. I beg to differ: how disagreement is handled in the annotation of legal machine learning data sets. *Artificial intelligence and law*, 32(3):839–862.

Amanda Cercas Curry, Gavin Abercrombie, and Verena Rieser. 2021. ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7388–7403, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Huang-Cheng Chou and Chi-Chun Lee. 2019. Every Rating Matters: Joint Learning of Subjective Labels and Individual Annotators for Speech Emotion Classification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5886–5890.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

H.M. Fayek, M. Lech, and L. Cavedon. 2016. Modeling subjectiveness in emotion recognition with deep neural networks: Ensembles vs soft labels. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 566–570.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond Black & White: Leveraging Annotator Disagreement via Soft-Label Multi-Task Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Benoit Frenay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. 2018. Who Said What: Modeling Individual Labelers Improves Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). Section: AAAI Technical Track: Machine Learning.

Heinrich Jiang and Ofir Nachum. 2020. Identifying and Correcting Label Bias in Machine Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 702–712. PMLR.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT understands, too. *AI Open*.

Yiwei Luo, Dallas Card, and Dan Jurafsky. 2020. DeSMOG: Detecting Stance in Media On Global Warming. *CoRR*, abs/2010.15149. ArXiv: 2010.15149.

Desmond Patton, Philipp Blandfort, William Frey, Michael Gaskell, and Svebor Karaman. 2019. Annotating Social Media Data From Vulnerable Populations: Evaluating Disagreement Between Domain Experts and Graduate Student Annotators. In *Proceedings of the 52nd Hawaii International Conference on System Sciences 2019 (HICSS-52)*.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent Disagreements in Human Textual Inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Barbara Plank. 2022. The “problem” of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Massimo Poesio. 2020. Ambiguity. In *The Wiley Blackwell Companion to Semantics*, pages 1–38. John Wiley & Sons, Ltd. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118788516.sem098.

Fabio Poletto, Valerio Basile, Cristina Bosco, Viviana Patti, Marco Stranisci, and others. 2019. Annotating hate speech: Three schemes at comparison. In *CEUR WORKSHOP PROCEEDINGS*, volume 2481, pages 1–8. CEUR-WS.

Vinodkumar Prabhakaran, Aida Mostafazadeh Davani, and Mark Diaz. 2021. On releasing annotator-level labels and information in datasets. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 133–138, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timothy RC Read and Noel AC Cressie. 2012. *Goodness-of-fit statistics for discrete multivariate data*. Springer Science & Business Media.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. 2008. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1):157–173.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *LREC*, pages 859–866.

Sanjana Sharma, Saksham Agrawal, and Manish Shrivastava. 2018. Degree based classification of harmful speech using Twitter data. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 106–112, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nishant Subramani, Samuel Bowman, and Kyunghyun Cho. 2019. Can Unconditional Language Models Recover Arbitrary Sentences? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob-Junqi Tian, David B. Emerson, Sevil Zanjani Miyandoab, Deval A. Pandya, Laleh Seyyed-Kalantari, and Faiza Khan Khattak. 2023. Soft-prompt Tuning for Large Language Models to Evaluate Bias. *ArXiv*, abs/2306.04735.

Cagri Toraman, Furkan Şahinuç, and Eyup Yilmaz. 2022. Large-Scale Hate Speech Detection with

Cross-Domain Transfer. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2215–2225, Marseille, France. European Language Resources Association.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from Disagreement: A Survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. 2023. Everyone's Voice Matters: Quantifying Annotation Disagreement Using Demographic Information. In *AAAI Conference on Artificial Intelligence*.

Dennis L. Wilson. 1972. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421.

Native language Identification for Arabic Language Learners using Pre-trained Language Models

Mohamed Amine Cheragui[1], **Mourad Abbas** [2] and **Mohammed Mediani**[3]
[1] Mathematics and Computer Science Department Ahmed Draia University, Adrar - Algeria
[2] High Council of Arabic - Algeria
[3] College of Information Technology, United Arab Emirates University Al-Ain, UAE
`m_cheragui@univ-adrar.edu.dz`
`m_abbas04@yahoo.fr`
`mohammed.mediani@uaeu.ac.ae`

Abstract

In this paper, we conduct an empirical study designed to systematically evaluate the efficacy of deep learning approaches in Native Language Identification (NLI) for native and non-native Arabic speakers. Specifically, we utilize three models: CAMeLBERT, AraBERTv0.2, and ARBERTv2. Our analysis is structured around two classification scenarios: binary classification and multi-class classification. This methodological framework allows us to comprehensively assess the performance of each model for the designated task.

1 Introduction

Native Language Identification is a specialized area within natural language processing (NLP) focused on automatically determining an individual's first language (L1) or mother tongue based on their written or spoken text in a second language (L2). This field involves the analysis of various linguistic features—including vocabulary usage, syntax, and stylistic patterns—to deduce the most likely native language of a writer or speaker. This process is predicated on the hypothesis that linguistic characteristics of the mother tongue often manifest in the acquisition and use of a second language, a phenomenon known as language transfer (Zampieri et al., 2017).

NLI offers a range of practical applications across diverse fields: Authorship Identification (Authorship Attribution) (Jarvis and Paquot, 2015), Author Profiling (Estival et al., 2007), Forensic Linguistics, (Mohammadi et al., 2017), Human-machine voice interface applications (Qian et al., 2017), Second Language Acquisition (SLA) (Malmasi and Dras, 2017b), Educational Technology Development (Laufer and Girsai, 2008), Marketing (Chen et al., 2017), and Security (Malmasi and Dras, 2017a).

In the literature, most research on NLI has focused on integrating linguistic features with machine learning methods (Tetreault et al., 2013). Key linguistic features analyzed include part-of-speech (POS) tagging (Gebre et al., 2013), character n-grams (Kulmizev et al., 2017), spelling errors (Kyle et al., 2015), and syntactic features (Wong and Dras, 2011). Commonly employed machine learning techniques in this domain include Naïve Bayes (NB) and Support Vector Machines (SVM). This combination leverages both detailed linguistic analysis and advanced computational models to effectively predict the native language of individuals from their second language texts.

The objective of our study is to conduct a series of experiments to investigate the efficacy of deep learning approaches in NLI for Arabic language learners. We explore this through two classification scenarios: binary classification and multi-class classification. To this end, we employ three models based on Bidirectional Encoder Representations from Transformers (BERT): CAMeLBERT (Inoue et al., 2021), AraBERTv0.2 (Antoun et al., 2020), and ARBERTv2 (Abdul-Mageed et al., 2021). These models are specifically implemented to assess the contribution of deep learning techniques in accurately identifying the native languages of Arabic language learners.

The structure of this paper is organized as follows: Section 2 reviews related work in NLI, offering background and context for our study. Section 3 describes the methodology and datasets used in our experiments, detailing the computational models, analysis techniques, and evaluation of each model's performance across various classification scenarios. Section 4 discusses the findings. Finally, Section 5 concludes the paper and suggests potential directions for future research in this field.

2 Literature Review

Like all other topics specific to natural language processing, research in NLI was focused essentially on learning English. However, in recent years a number of studies have focused on other languages as Chinese, Norwegian, Portuguese and Arabic.

2.1 English Learning Language

(Tetreault et al., 2012) conducted a pioneering study on the use of classifier ensembles for NLI. The study employed an ensemble of logistic regression learners, utilizing a diverse set of features including character and word n-grams, function words, parts of speech, spelling errors, and writing quality markers. For syntactic features, they explored the use of Tree Substitution Grammars and dependency features obtained using the Stanford parser. They also proposed incorporating language models into NLI and used language model perplexity scores based on lexical 5-grams from each language in their corpus. The ensemble model achieved accuracies, with 90.1% on the ICLE (Granger et al., 2009) and 80.9% on the TOEFL11 corpus (Blanchard et al., 2013), respectively.

(Lotfi et al., 2020) proposed a deep generative language modelling (LM) approach to NLI. Their approach is to fine-tune a GPT-2 model separately on texts written by the authors with the same L1, and assigning a label to an unseen text based on the minimum LM loss with respect to one of these fine-tuned GPT-2 models. They evaluated their approach using two datasets, TOEFL11 and ICLE, achieving an accuracy of 86.6% and 94.2% respectively.

(Uluslu and Schneider, 2022) described ProDAPT, transformer adapters based on deep generative model, which is considered as an alternative lightweight fine-tuning strategy that achieves equal performance to full fine-tuning on most tasks. In terms of performance, their model achieved 82.4% accuracy on TOEFL11 corpus.

2.2 Arabic Learning Language

(Malmasi and Dras, 2014) presented the first application for NLI to Arabic learners, based on a supervised multi-class classification approach, by combining three syntactic features (CFG production rules, Arabic function words and Part-of-Speech n-grams. To perform multi-class classification, they used SVM. The system achieves an accuracy of 41% on ALC Corpus.

(Mechti et al., 2020) studied the impact of automatic classification using some data statistically extracted from a source corpus, to detect the mother tongue of Arabic learners. They combined three syntactic features which are: Part of speech n-grams, function words and context-free grammar production rules. For the classification, the LIBSVM2 was used, as variant of SVM. For training and evaluation, they opted for Arabic Learner Corpus, in which their model obtained an accuracy of 45%.

(Ionescu, 2015) presented a study based on a machine learning method that works at the character level, using a kernel based on Local Rank Distance (LRD). The resulting model of this combination was trained and tested on ALC, obtained an accuracy score of 50.1%.

2.3 Other Learning Languages

(Malmasi et al., 2015) proposed NLI experiments on Norwegian language, by employing a supervised multi-class classification approach, which takes into consideration three syntactic feature types: function Words, part-of-Speech n-grams and mixed POS-function word n-grams. As a dataset for training and evaluation they used the ASK Corpus (Tenfjord et al., 2006). The model achieved an accuracy score of 78.6%.

(Remnev, 2019) developed a model for Russian Native Language Identification, based on the support vector method and the TF-IDF metric. To train and evaluate the proposed model, he used the Russian Learner Corpus. In terms of performance, the adopted approach achieved an accuracy score of 80%. (Malmasi et al., 2018) presented a study about native Language Identification for learners of Portuguese (as L2 Language). The used approach is a combination of linguistic features and Machine Learning. The authors defined three features which are: Function words, Context-free grammar production rules and Part-of-Speech (POS) tags. They also utilized a standard multi-class classification approach, by using linear Support Vector Machines. For the dataset, they used NLI-PT (del Río Gayo et al., 2018). The proposed model attained an accuracy of 54.1% .

(del Río, 2020) investigated the impact of different linguistic features in NLI for L2 Portuguese. For that, she defined two types of lexical features: one includes all the words in the text, and the other one includes all the words except nouns and adjectives. In addition, other morphological and syntactic features have been used, including: POS, context-free grammar (CFG) production rules and dependency triplets. For the experiment, she used 04 classifiers,

which are: Multinomial Logistic Regression, SVM, Ridge Regression and Multi-Layer Perceptron classifier, which have been trained and tested on the NLI-PT dataset. In terms of performance, the MLP classifier achieved the best accuracy of 66%.
(Uluslu, 2023) presented an application of NLI specifically for Turkish language learners. The approach employed a combination of three syntactic features: Context-Free Grammar (CFG) production rules, part-of-speech n-grams, and function words. The study used a standard supervised multi-class classification method, where a linear Support Vector Machine (SVM) was applied for classification. Feature vectors were created using a TF-IDF weighting scheme. The Turkish Learner Corpus (TLC) (Golynskaia, 2022), was utilized to evaluate the system's performance. By combining the three features, the proposed system achieved an accuracy score of 44.2%.

3 Experimental Methodology

3.1 Data

For our experiment, we used the Arabic Learner Corpus (ALC) (Alfaifi et al., 2014). The Corpus has been used for various studies in language learning and computational linguistics focusing on Arabic. It comprises a collection of written and spoken materials produced by learners of Arabic, which are used for different types of linguistic research and language teaching tool development. The dataset was compiled during the years 2012 and 2013. It comprises 282,732 words and consists of 1585 texts, encompassing both written and spoken content. These texts were generated by a total of 942 students learning Arabic, representing 67 nationalities and originating from 66 distinct mother tongue backgrounds. In addition, ALC includes 26 variables as metadata elements, 12 for the learner and 14 for the text.

3.2 Models

The aim of our research is to examine the impact of various pre-trained Arabic BERT models by exploring different combinations of classification task related to native language identification.
To achieve this, we fine-tuned 03 models, including AraBERTV0.2, ARBERTv2 and CAMeLBERT Using Arabic Learner Corpus. Each model was used to execute 02 Scenarios, which are: Binary classification and Multi-class classification.
Our choice of these models was made for a number of reasons: they have been specifically pre-trained on large-scale Arabic corpora, which helps them capture the nuances and intricacies of Arabic. They have demonstrated competitive performance on various NLP tasks (Sentiment Analysis, Language Identification, Named Entity Recognition, Fake News Detection, etc). Their architectures and training procedures are designed to achieve state-of-the-art results on a range of Arabic language understanding tasks, making them suitable choices for classification tasks as well. These models often come in different version (Large/base) and variants (MSA/Dialect).

It's also important to mention that even though the 03 models were developed based on the same architecture (BERT), there are a number of distinguishing features.

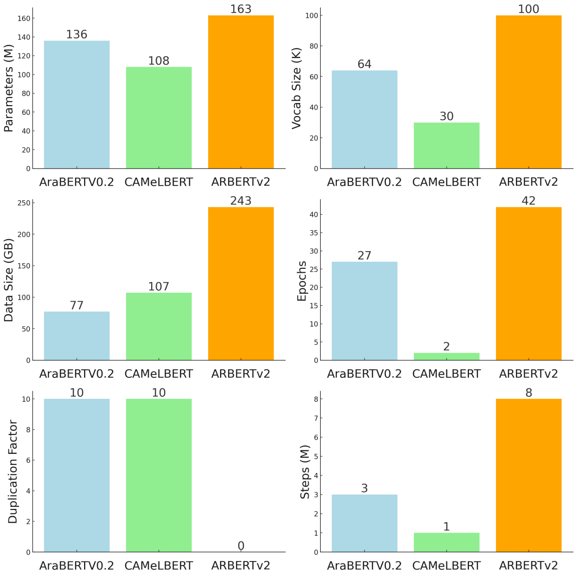


Figure 1: Configurations of used models.

Parameter	Value
Epochs	05
Batch	08
Learning rate	4.87 e-5
weight decay	0.01
seed	20

Table 1: Hyper-parameters values.

3.2.1 Binary Classification

Binary classification is a fundamental task in machine learning where the goal is to classify input data into one of two possible categories or classes (Er et al., 2016). To do this, We carried out two experiments, the first concerning the identification of Arabic as the mother tongue of learners, in order to fine-tuning our models, we have divided our dataset into two distinct categories. The first category is labelled "1", which concerns texts whose authors native language is Arabic. The rest of the texts constituting the second category will be labelled "0" (Table 2 summarises the different test results). The second experiment mirrors the first, but in this instance, we handle each of the six languages individually, applying the same process to each one (the result is given in the table 3).

3.2.2 Multi-class Classification

Multi-class classification involves classifying data into more than two groups/categories (Fields et al., 2024). Unlike binary classification, where the model is trained to predict only one of the two classes of an item, a multi-class classifier is trained to predict one of the three or more classes of an item. In our case, we set up two experiments, the first for detecting Arabic language learners' mother tongues and the second dealing with the same task based on level of study.

- **Multi-class classification for detecting Arabic learners' mother tongue**: The corpus comprises 66 distinct mother tongue representations. However, the number of representative texts varies from one Mother tongue to another, so we kept only languages with a good quantitative representation in terms of texts, as shown in figure 2. The results of this experiment are given in table 3.

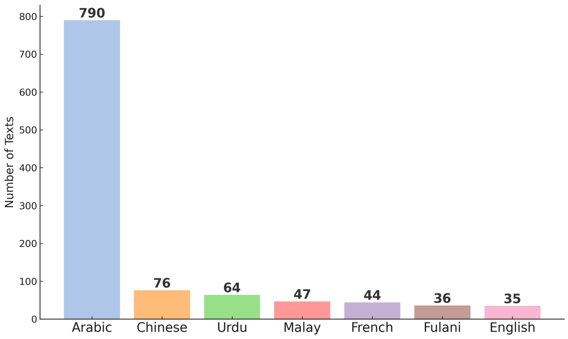


Figure 2: Number of texts produced by Arabic language learners with a mother tongue other than Arabic (ALC corpus).

- **Multi-class classification based on Level of Study**: The ALC contains 05 categories of learners according to their level of study: secondary school, general language course, diploma programme (advanced language course), Bachelor degree and Master degree. Learners of both the Bachelor degree and Master degree were majoring in Arabic. Figure 3 gives an estimate of the percentage of each level of study in the ALC. For results, table 5 presents a global view of the performance of three models in detecting the mother tongue based on levels of study, offering a comparative understanding of their effectiveness in this classification task. The Table 6 offers nuanced insights into its ability of the CAMeLBERT model to capture specific mother tongue differences at each level, providing a more refined understanding of its classification precision in this context.

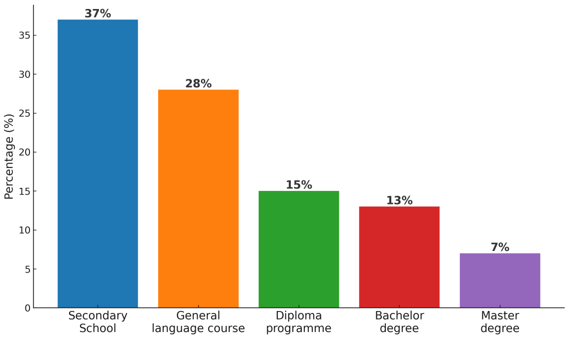


Figure 3: Corpus distribution by Level of Study.

4 Discussion

After reviewing the results, we found that the three models achieved good results for binary classification, but in the Multi-class classification there was a significant decrease in the effectiveness of the models, which is probably due to the fact that deep learning models such as CAMeLBERT, AraBERTv0.2 and ARBERTv2 have differences in performance when applied to multi-class classification task due to several reasons:

- Model structure and training data: These models, being variants of BERT (bi-directional encoding representations of transforms), are primarily designed to capture complex patterns in text through deep bi-directional representations. However, the effectiveness of these models is highly dependent on the quality and diversity of the training data. For Arabic with many dialects and a rich morphological

Metrics	CAMeLBERT	AraBERTv0.2	ARBERTv2
Accuracy	97.71%	97.26%	96.34%
Precision	96.51%	95.62%	94.37%
Recall	97.14%	96.84%	95.57%
F1	96.82%	96.21%	94.95%

Table 2: Binary Classification, One-versus-All (Arabic versus the six other languages).

Metrics	CAMeLBERT	AraBERTv0.2	ARBERTv2
Chinese	98.32%	96.14%	94.43%
Urdu	97.66%	95.89%	95.09%
Malay	96.15%	97.23%	95.77%
French	97.73%	98.02%	96.41%
Fulani	98.18%	97.86%	95.31%
English	97.06%	96.23%	93.86%

Table 3: Binary Classification One-versus-One (Arabic/Non-Arabic).

Metrics	CAMeLBERT	AraBERTv0.2	ARBERTv2
Accuracy	87.21%	83.10%	81.27%
Precision	64.74%	33.49%	28.67%
Recall	61.74%	40.59%	35.91%
F1	60.43%	36.45%	30.58%

Table 4: Multi-class detection of mother tongue learners'.

Metrics	CAMeLBERT	AraBERTv0.2	ARBERTv2
Accuracy	80.82%	75.79%	74.42%
Precision	80.00%	77.74%	62.92%
Recall	63.22%	50.02%	48.41%
F1	66.13%	53.55%	50.86%

Table 5: Global view on multi-class classification performance based on Level of study using CAMeLBERT, AraBERTv0.2 and ARBERTv2.

Metrics	Precision	Recall	F1 Score
Secondary school	58.64%	40.44%	48.45%
General language course	60.08%	45.58%	52.19%
Diploma programme	62.12%	49.24%	56.78%
Bachelor degree	78.45%	54.97%	63.32%
Master degree	80.25%	57.48%	66.45%

Table 6: Detailed scores for multi-class classification based on Level of study using CAMeLBERT.

structure, models trained on Standard Arabic may not perform well when faced with dialect-related variations unless they are specifically tuned to diverse datasets that include such variations.

- Task complexity: Multi-class classification task are inherently more complex than binary classification. In multi-class classification, the model must choose the correct class among several possible classes, which increases the chance of error, especially if some classes are underrepresented in the training data. Multi-class classification adds another layer of complexity since each sample may belong to multiple classes simultaneously, requiring the model to understand and predict all applicable classes.

- Imbalance between categories: Often, in multi-category settings, some categories contain far more examples than others. This imbalance can lead to models that are biased towards more frequent categories, reducing their overall effectiveness across less frequent categories. Similarly, in multi-class settings, some classes may be repeated more frequently than others, which can skew the model's predictions.

- Fine-tuning and adaptation: While models like AraBERTv0.2, CAMeLBERT, and ARBERTv2 are pre-trained on a large set of models, their performance on specific tasks such as multi-class classification or multi-label classification can depend on how well they are tuned. Fine-tuning on a task-specific dataset is critical, but without sufficient task-specific data or proper organization, models can over-adapt to the training data and perform poorly on unseen data.

- Linguistic nuances: Arabic language processing poses unique challenges due to the richness of the Arabic language in terms of linguistic form and the presence of many homographs (words that are spelled the same way but have different meanings). Effective processing of these nuances requires either specialized pre-processing or structures designed to better capture these aspects, which can be a limitation of general-purpose models such as AraBERTv0.2, CAMeLBERT, and ARBERTv2 when they are not modified for such details.

5 Conclusion

In this paper, we conducted a comparative study of deep learning models for a classification task using the Arabic Language Learners' Corpus (ALC). We evaluated three models based on the BERT architecture: CAMeLBERT, AraBERTv0.2, and ARBERTv2. These models were fine-tuned and tested on two classification scenarios: binary and multi-class.

The experimental results indicate that all three models perform exceptionally well in binary classification, with F1 scores of 96.82% for CAMeLBERT, 96.21% for AraBERTv0.2, and 94.95% for ARBERTv2. However, the performance decreased for multi-class classification. CAMeLBERT achieved the highest performance in both subcategories: 60.43% for categorization based on mother tongue and 66.13% for that based on school level. In contrast, the F1 score related to the other two models did not exceed 37% for the first subcategory and 54% for the second one.

The noticeable decrease in performance of the three models in the multi-class classification task can be attributed to two main factors: firstly, the size of the corpus used and the disparities in the number of texts between languages and grade levels; and secondly, the increased complexity of these classifications compared to binary classification.

Comparing the three models, we found that CAMeLBERT's outperforms ARBERTv2 and AraBERTv0.2. This can be ascribed to several parameters: an extensive and diverse training corpus, effective fine-tuning of tasks, architectural innovations, and robust benchmark results.

For future work, we plan to incorporate additional linguistic features such as syntactic and Part of speech tagging to enhance the models' efficiency.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Abdullah Alfaifi, Eric Atwell, and Hedaya Ibraheem. 2014. Arabic learner corpus (alc) v2: a new written and spoken corpus of arabic learners. In *International Symposium Learner Corpus Studies in Asia and the World (LCSAW)*, volume 2, pages 77 – 89. Kobe International Communication Center.

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *CoRR*, abs/2003.00104.

Daniel Blanchard, Joel R. Tetreault, Derrick Higgins, A. Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013:15.

Lingzhen Chen, Carlo Strapparava, and Vivi Nastase. 2017. Improving native language identification by using spelling errors. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 542–546, Vancouver, Canada. Association for Computational Linguistics.

Iria del Río. 2020. Native language identification on l2 portuguese. In *Computational Processing of the Portuguese Language*, pages 87–97, Cham. Springer International Publishing.

Iria del Río Gayo, Marcos Zampieri, and Shervin Malmasi. 2018. A Portuguese native language identification dataset. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Meng Joo Er, Rajasekar Venkatesan, and Ning Wang. 2016. An online universal classifier for binary, multi-class and multi-label classification. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003701–003706.

Dominique Estival, Tanja Gaustad, Son Bao Pham, and Will Radford. 2007. Profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 263, page 272.

John Fields, Kevin Chovanec, and Praveen Madiraju. 2024. A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*, 12:6518–6531.

Binyam Gebrekidan Gebre, Marcos Zampieri, Peter Wittenburg, and Tom Heskes. 2013. Improving native language identification with TF-IDF weighting. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 216–223, Atlanta, Georgia. Association for Computational Linguistics.

Anna Golynskaia. 2022. An error coding system for the turkish learner corpus. *The Journal of Linguistics*, 0(39):67–87.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, Magali Paquot, et al. 2009. *International corpus of learner English*, volume 2. Presses universitaires de Louvain Louvain-la-Neuve.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *CoRR*, abs/2103.06678.

Radu Tudor Ionescu. 2015. A fast algorithm for local rank distance: Application to arabic native language identification. In *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part II 22*, pages 390–400. Springer.

Scott Jarvis and Magali Paquot. 2015. *Learner corpora and native language identification*, Cambridge Handbooks in Language and Linguistics, page 605–628. Cambridge University Press.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Copenhagen, Denmark. Association for Computational Linguistics.

Kristopher Kyle, Scott Andrew Crossley, and Youjin Kim. 2015. Native language identification and writing proficiency.

Batia Laufer and Nany Girsai. 2008. Form-focused Instruction in Second Language Vocabulary Learning: A Case for Contrastive Analysis and Translation. *Applied Linguistics*, 29(4):694–716.

Ehsan Lotfi, Ilia Markov, and Walter Daelemans. 2020. A deep generative approach to native language identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1778–1783, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shervin Malmasi, Iria del Río, and Marcos Zampieri. 2018. Portuguese native language identification. In *Computational Processing of the Portuguese Language: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018, Proceedings 13*, pages 115–124. Springer.

Shervin Malmasi and Mark Dras. 2014. Arabic native language identification. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 180–186, Doha, Qatar. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2017a. Multilingual native language identification. *Natural Language Engineering*, 23(2):163–215.

Shervin Malmasi and Mark Dras. 2017b. Native language identification using stacked generalization. *arXiv preprint arXiv:1703.06541*.

Shervin Malmasi, Mark Dras, and Irina Temnikova. 2015. Norwegian native language identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 404–412, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Seifeddine Mechti, Nabil Khoufi, and Lamia Hadrich Belguith. 2020. Improving native language identification model with syntactic features: Case of arabic. In *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 2*, pages 202–211. Springer.

Elham Mohammadi, Hadi Veisi, and Hessam Amini. 2017. Native language identification using a mixture of character and word n-grams. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 210–216, Copenhagen, Denmark. Association for Computational Linguistics.

Yao Qian, Keelan Evanini, Xinhao Wang, David Suendermann-Oeft, Robert A. Pugh, Patrick L. Lange, Hillary R. Molloy, and Frank K. Soong. 2017. Improving Sub-Phone Modeling for Better Native Language Identification with Non-Native English Speech. In *Proc. Interspeech 2017*, pages 2586–2590.

Nikita Remnev. 2019. Native language identification for russian. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 1–7.

Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. The ask corpus–a language learner corpus of norwegian as a second language.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta, Georgia. Association for Computational Linguistics.

Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In *Proceedings of COLING 2012*, pages 2585–2602, Mumbai, India. The COLING 2012 Organizing Committee.

Ahmet Yavuz Uluslu. 2023. Turkish native language identification. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 303–307, Online. Association for Computational Linguistics.

Ahmet Yavuz Uluslu and Gerold Schneider. 2022. Scaling native language identification with transformer adapters. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 298–302, Trento, Italy. Association for Computational Linguistics.

Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting parse structures for native language identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Marcos Zampieri, Alina Maria Ciobanu, and Liviu P Dinu. 2017. Native language identification on text and speech. *arXiv preprint arXiv:1707.07182*.

Generative Adversarial Network based Neural Vocoder for Myanmar End-to-End Speech Synthesis

Aye Mya Hlaing, Win Pa Pa
Natural Language Processing Lab.,
University of Computer Studies, Yangon, Myanmar
`{ayemyahlaing, winpapa}@ucsy.edu.mm`

Abstract

Recently, neural vocoders have been employed in end-to-end speech synthesis for converting the intermediate spectral representations to the corresponding speech waveform. In this paper, two generative adversarial network (GAN) based vocoders, Parallel WaveGAN and HiFi-GAN are proposed for Myanmar end-to-end speech synthesis and subjective evaluations are conducted to compare the performance of the models. The subjective evaluation results show that both models trained on small Myanmar speech dataset achieve the high fidelity speech synthesis with fast inference speed, showing the ability of generalizing to the mel-spectrogram inversion of unseen speakers. Specifically, in end-to-end speech synthesis, Tacotron2 with HiFi-GAN vocoder achieves state-of-the-art performance resulting in a 4.37 mean opinion score (MOS) for Myanmar language.

1 Introduction

Text-to-speech (TTS) models focus on synthesizing intelligible and natural sounding speech which are indistinguishable from the original human speech. For the past few decades, statistical parametric speech synthesis (SPSS) has been the dominant technology for TTS (Tokuda et al., 2013; Qian et al., 2014; Wu et al., 2015; Zen and Sak, 2015). However, SPSS needs a complex pipeline for getting language dependent good linguistic features and that is time consuming and very expensive. This paper is a part of the ASEAN IVO 2023 project, "Spoof Detection for Automatic Speaker Verification", which aims to enhance the security and reliability of speaker verification by effectively detecting spoofing attacks.

In recent years, end-to-end neural TTS models, such as Tacotron (Wang et al., 2017), Tacotron2 (Shen et al., 2018), Transformer based TTS (Li et al., 2019), FastSpeech (Ren et al., 2019), FastSpeech2 (Ren et al., 2020), have emerged to simplify traditional speech synthesis pipeline and their synthesized speeches can be comparable with human recordings. The end-to-end neural TTS is typically composed of two main processing models, the spectral representation generator and the vocoder. The first one generates the spectral representation such as mel-spectrograms given the input text or phoneme, and the vocoder converts the speech waveforms from the generated mel-spectrograms. Griffin Lim algorithm (Griffin and Lim, 1984), the classic phase estimation method is generally used for speech waveform reconstruction.

Recently, in the context of end-to-end TTS synthesis, the separately trained neural vocoders such as WaveNet (Van Den Oord et al., 2016), Parallel WaveNet (Oord et al., 2018), MelGAN (Kumar et al., 2019), WaveGlow (Prenger et al., 2019), Parallel WaveGAN (Yamamoto et al., 2020) and HiFi-GAN (Kong et al., 2020) have demonstrated remarkable capabilities in generating natural-sounding synthetic speech. Inspired by this factor, in this work, the advantage of neural vocoder is combined into the Myanmar end-to-end speech synthesis to achieve both efficient and high-fidelity speech synthesis.

We trained two generative adversarial network based neural vocoders, Parallel WaveGAN and HiFi-GAN on Myanmar speech dataset because of their remarkable performance on generating waveform at fast inference speed while maintaining the quality of speech comparative to the other neural vocoders. To confirm the effectiveness of the vocoders, experiments were conducted by utilizing them in different conditions. We examined the ability of each vocoder in ground truth mel-spectrogram inversion, generalization on unseen speakers, and Myanmar end-to-end speech synthesis. Audio samples are available on this website[1].

---

[1] http://nlpresearch-ucsy.edu.mm/subeval-voc.html

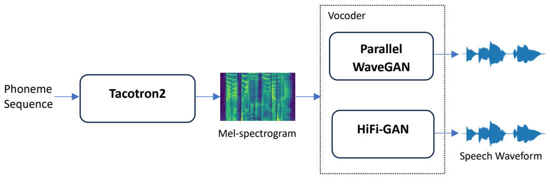


Figure 1: A proposed model of End-to-End TTS with GAN-based neural vocoders

2 Myanmar End-to-End Speech Synthesis

For Myanmar language, statistical parametric speech synthesis with different input linguistic features have been done on Myanmar speech synthesis. In HMM-based Myanmar TTS (Thu et al., 2015), CART-based Myanmar TTS (Hlaing and Pa, 2018), DNN-based Myanmar speech synthesis (Hlaing et al., 2018), LSTM-RNN-based Myanmar speech synthesis (Hlaing et al., 2019; Hlaing and Pa, 2020, Oo et al., 2020), we found that traditional speech synthesis pipeline and traditional vocoder such as WORLD vocoder (Morise et al., 2016) were used.

The first end-to-end Myanmar TTS System based on Tacotron was introduced in (Win et al., 2020) and Tacotron2 based end-to-end Myanmar TTS with phone-level embedding was found in (Qin et al., 2020). However, there is no research on the effectiveness of neural vocoder specifically trained on Myanmar speech dataset for Myanmar end-to-end TTS. To the best of our knowledge, this is the first effort to explore the advance of neural vocoder in Myanmar end-to-end TTS.

Figure 1 shows our proposed model of Myanmar end-to-end TTS with generative adversarial network based neural vocoders. In this work, a Tacotron2 (Shen et al., 2018) model was trained for the purpose of text to mel-spectrogram generation and the generated mel-spectrograms were given into our separately trained GAN-based vocoders, including Parallel WaveGAN and HiFi GAN as the input conditions. Tacotron2 uses character sequences as input, but our model was trained on phoneme sequences to alleviate the mispronunciation problems of rarely occurred words in the small training set.

3 GAN based Neural Vocoder

The first attempt of applying GAN (Goodfellow et al., 2014) to the synthesis of raw-waveform audio is WaveGAN (Donahue et al., 2018) and followed by many variants of GAN-based vocoders such as MelGAN (Kumar et al., 2019), StyleMelGAN (Mustafa et al., 2021), Multi-band MelGAN (Yang et al., 2021), Parallel WaveGAN (Yamamoto et al., 2020) and HiFi-GAN (Kong et al., 2020). GAN-based vocoders show significant performance over autoregressive models in the speed and quality of synthesized speech (AlBadawy et al., 2022). Among the different variants of GAN-based vocoders, we selected to train the vocoders using Parallel WaveGAN and HiFi-GAN for Myanmar end-to-end speech synthesis.

3.1 Parallel WaveGAN

The Parallel WaveGAN (Yamamoto et al., 2020) is a distillation-free, fast, and small-footprint waveform generation method using GAN. Though a WaveNet-based model conditioned on mel-spectrogram is used as the generator, the model is non-autoregressive at both training and inferencing. The generator is trained by jointly optimizing the multi-resolution short-time Fourier transform (STFT) auxiliary loss L_{aux} and the waveform-domain adversarial loss L_{adv}.

$$L_G = L_{aux}(G) + \lambda_{adv} L_{adv}(G, D) \tag{1}$$

where λ_{adv} represents the hyperparameter that balances the two loss terms.

Meanwhile, the discriminator is trained to correctly classify the generated sample as fake and simutaneously ground truth sample as real with the following equation:

$$L_D = \mathbb{E}_{x \sim p}[(1 - D(x))^2] + \mathbb{E}_z[D(G(z))^2] \tag{2}$$

where x denotes the target waveform, p denotes its distribution, and z denotes the input white noise.

3.2 HiFi-GAN

HiFi-GAN has been composed of one generator and two discriminators containing multi-scale discriminator (MSD) and multi-period discriminator (MPD) (Kong et al., 2020). The generator of HiFi-GAN is a fully convolutional neural network with multi-receptive field fusion (MRF) module that can perceives the various length of patterns in parallel. The final loss terms for the generator in HiFi GAN is as follows:

$$L_G = L_{Adv}(G; D) + \lambda_f L_F(G; D) + \lambda_m L_M(G) \tag{3}$$

where L_F and L_M are the feature matching loss and mel-spectrogram loss, respectively.

In the discriminator part, each sub-discriminator of MPD handles equally spaced samples of input audio and MSD was used to capture consecutive patterns and long-term dependencies. The discriminator with respect to the sub-discriminators of MPD and MSD is as follows:

$$L_D = \sum_{k=1}^{K} L_{Adv}(D_k; G) \tag{4}$$

where D_k represents k-th sub-discriminator in MPD and MSD.

4 Experiments

The dataset and the experimental setups of the models are presented in this section. The training of both GAN-based vocoders had been conducted on the open-source implementation from this site[2] and Tacotron2 model was implemented using ESPnet[3], an end-to-end speech processing toolkit. Each vocoder was trained on a single Nvidia Tesla K80 GPU and Tacotron2 model was trained on two Nvidia Tesla K80 GPUs.

4.1 Dataset

For training our proposed end-to-end pipeline including Tacotron2 model, Parallel WaveGAN and HiFi-GAN vocoders, we used a Myanmar phonetically balanced speech corpus (PBC) (Thu et al., 2015) built from Basic Travel Expression Corpus (BTEC) (Kikui et al., 2003) recorded by a female native speaker. In total, 3,800 utterances were utilized for training, 100 utterances each for validation and testing. The sampling rate of speech data was 16kHz.

4.2 Experimental setup of Parallel WaveGAN

For training the Parallel WaveGAN on Myanmar speech dataset, we used 80-band log-mel spectrograms with band-limited frequency range (80 to 7600 Hz) as the input auxiliary features for waveform generation models. The same configuration setting for the generator and the discriminator networks with the original paper (Yamamoto et al., 2020) was used in our work. Weight normalization was applied to all convolutional layers of both generator and discriminator. The hyperparameter λ_{adv} in Equation 1 was also set to 4.0. The model was trained for 200K steps and the discriminator was fixed for the first 100K steps, and then both the generator and the discriminator were trained jointly. We set the length of each audio clip to 25600 and mini-batch size to 6. The generator was set with the initial learning rate of 1×10^{-4} and the discriminator with the initial learning rate of 5×10^{-5}.

4.3 Experimental setup of HiFi-GAN

Among the variations of the generators in original source of HiFi-GAN(Kong et al., 2020), the configuration of HiFi-GAN V1 was applied to train the model on Myanmar speech dataset. We used 80-band log-mel spectrograms with band-limited frequency range (80 to 7600 Hz) as input conditions. The FFT and hop size were set to 1024 and 256, respectively. Adam (Kingma and Ba, 2014) optimizer with $\beta_1 = 0.5, \beta_2 = 0.9$ was used for training both the generator and the discriminator networks, and the initial learning rate was set to 2×10^{-4}. The batch size was 16 and the length of each audio clip was 8192. The model was trained for only 200K steps, the same steps used for training the Parallel WaveGAN model. This is very small compared to the training steps used in the original paper (2.5M steps).

4.4 Experimental setup of Tacotron2

Tacotron2 (Shen et al., 2018), a recurrent sequence-to-sequence feature prediction network with attention that maps phoneme embeddings to mel-spectrograms, was trained on the dataset mentioned in section 4.1 with a batch size of 32. The model was trained for 125K steps with Adam optimizer (Kingma and Ba, 2014) and a learning rate of 1×10^{-3}. In the training process, the guided attention loss was used to promote a fast and robust attention learning.

5 Results

To examine the performance of our trained Parallel WaveGAN and HiFi-GAN models, three mean opinion score (MOS) tests were performed for ground truth mel-spectrogram inversion, mel-spectrogram inversion for unseen speakers, and end-to-end Myanmar speech synthesis tasks. Ten native non-expert speakers participated in all MOS tests. Subjects were given the synthesized speeches of two models and ground truth audio, and they had to rate the quality of synthesized speeches on a scale of 1 to 5 where 1 is bad and 5 is excellent. The speech samples were randomly ordered.

---

[2] https://github.com/kan-bayashi/ParallelWaveGAN

[3] https://github.com/espnet/espnet

Model	MOS	RTF
Ground Truth	4.69 ± 0.10	-
Parallel WaveGAN	4.49 ± 0.12	0.015
HiFi-GAN	4.59 ± 0.11	0.011

Table 1: Comparison of MOS with 95% confidence intervals and inference speed (RTF) in ground truth mel-spectrogram inversion. Note that RTF is based on the average inference time of 100 utterances in evaluation set on a single Nvidia Tesla K80 GPU.

5.1 Ground Truth Mel-spectrogram Inversion

The MOS test and speed measurement with Real Time Factor (RTF) were performed to evaluate the performance of Parallel WaveGAN and HiFi-GAN models in terms of the quality of synthesized speeches and the inference speed. 10 utterances randomly selected from the testing set, were used for MOS test of mel-spectrogram inversion and the results are shown in Table 1. It can be clearly seen that both models can synthesize the high quality speech comparable to the ground truth speech. Remarkably, HiFi-GAN model achieves the highest MOS score 4.59 with a gap of 0.10 compared to the ground truth audio score 4.69 and this means that the synthesized speech is almost indistinguishable from the original speech. The RTF results indicate that both models get very small RTF values. Specifically, HiFi-GAN model gives the lowest RTF value (0.011) which implies that the model can synthesize speech 99.9 times faster than real-time on single Nvidia Tesla K80 GPU.

5.2 Generalization to Unseen Speakers

In this MOS test, 10 utterances of two unseen female speakers were utilized for investigating the ability of our trained models on generalizing to unseen speakers. However, we did not conduct an additional training for each model on multi-speaker dataset for this task. The MOS results for the mel-spectrogram inversion of the unseen speakers are shown in Table 2. The results show that Parallel WaveGAN and HiFi-GAN achieved 4.42 and 4.48 scores, respectively, indicating that both models can generalize well to unseen speakers.

5.3 End-to-End TTS

To verify the effectiveness of the Parallel WaveGAN and HiFi-GAN models in Myanmar end-to-end TTS pipeline, each model was integrated to the Tacotron2 model mentioned in section 4.4 as the vocoder. In the inferencing step, the Tacotron2 model convert the input phoneme sequences to the corresponding mel-spectrograms, and by inputting generated mel-spectrograms to vocoder models, they generate the corresponding speech waveform. To evaluate the quality of the generated speech samples, we conducted MOS test and the results are presented in Table 3. It can be observed that end-to-end TTS systems with independently trained neural vocoders can generate high quality synthesized speech. In particular, our model using Tacotron2 with Parallel WaveGAN vocoder achieves 4.33 MOS score which is comparable to the MOS results of the Parallel WaveGAN with the Transformer-based TTS (Yamamoto et al., 2020), and also the model using Tacotron2 with HiFi-GAN vocoder achieves 4.37 MOS score which is comparable to HiFi-GAN V1 model without fine-tuning (Kong et al., 2020) in the end-to-end TTS settings.

Model	MOS
Ground Truth	4.68 ± 0.12
Parallel WaveGAN	4.42 ± 0.12
HiFi-GAN	4.48 ± 0.11

Table 2: Comparison of MOS with 95% confidence intervals for generalizing on unseen speakers

Model	MOS
Ground Truth	4.68 ± 0.15
Tacotron2 + Parallel WaveGAN	4.33 ± 0.13
Tacotron2 + HiFi-GAN	4.37 ± 0.13

Table 3: Comparison of MOS with 95% confidence intervals in end-to-end Myanmar speech synthesis with neural vocoders

6 Conclusion

In conclusion, both Parallel WaveGAN and HiFi-GAN models achieve high-fidelity speech synthesis with fast inference speeds, showing the ability of generalizing to unseen speakers. By integrating these GAN-based models with Tacotron2 in the end-to-end TTS framework as the vocoders, we achieved the state-of-the-art speech quality for Myanmar language. Our work demonstrates that the GAN-based models, even trained on the small dataset with limited training steps, can achieve high quality speech for low-resource languages. Future work includes improving the mel-spectogram generator to better capture the prosody of speech and using GAN-based vocoders in various end-to-end speech synthesis settings.

Limitations

Due to the limited GPU resources, we can use the limited training steps on the models, however, more robustness of the models can be achieved by finetuning the hyperparemeters and training the models until an optimal point is reached. When the ability of vocoder is examined with the aim of generalizing to unseen speakers, one of the limitations is the unavailability of multi-speaker Myanmar dataset.

Acknowledgements

The ASEAN IVO (`http://www.nict.go.jp/en/asean_ivo/index.html`) project, "Spoof Detection for Automatic Speaker Verification", was involved in the production of the contents of this work and financially supported by NICT (`http://www.nict.go.jp/en/index.html`).

References

Ehab A AlBadawy, Andrew Gibiansky, Qing He, Jilong Wu, Ming-Ching Chang, and Siwei Lyu. 2022. Vocbench: A neural vocoder benchmark for speech synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 881–885. IEEE.

Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Daniel Griffin and Jae Lim. 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243.

Aye Mya Hlaing and Win Pa Pa. 2018. Word-based myanmar text-to-speech with clustergen. In *The 16th International Conference on Computer Applications (ICCA2018)*, pages 203–208.

Aye Mya Hlaing and Win Pa Pa. 2020. Word representations for neural network based myanmar text-to-speech s. *International Journal of Intelligent Engineering & Systems*, 13(2).

Aye Mya Hlaing, Win Pa Pa, and Ye Kyaw Thu. 2018. Dnn based myanmar speech synthesis. In *SLTU*, pages 142–146.

Aye Mya Hlaing, Win Pa Pa, and Ye Kyaw Thu. 2019. Enhancing myanmar speech synthesis with linguistic information and lstm-rnn. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 189–193.

Gen-ichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. 2003. Creating corpora for speech-to-speech translation. In *INTERSPEECH*, pages 381–384.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. 2016. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884.

Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs. 2021. Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Yin May Oo, Theeraphol Wattanavekin, Chenfang Li, Pasindu De Silva, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, Oddur Kjartansson, and Alexander Gutkin. 2020. Burmese speech corpus, finite-state text normalization and pronunciation grammars with an application to text-to-speech. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6328–6339.

Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Yao Qian, Yuchen Fan, Wenping Hu, and Frank K Soong. 2014. On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3829–3833. IEEE.

Qinglai Qin, Jian Yang, and Peiying Li. 2020. Myanmar text-to-speech synthesis using end-to-end model. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, pages 6–11.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in neural information processing systems*, 32.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Ye Kyaw Thu, Win Pa Pa, Jinfu Ni, Yoshinori Shiga, Andrew M Finch, Chiori Hori, Hisashi Kawai, and Eiichiro Sumita. 2015. Hmm based myanmar text to speech system. In *INTERSPEECH*, pages 2237–2241.

Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. 2013. Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, 101(5):1234–1252.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Yuzana Win, Htoo Pyae Lwin, and Tomonari Masada. 2020. Myanmar text-to-speech system based on tacotron (end-to-end generative model). In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 572–577. IEEE.

Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King. 2015. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4460–4464. IEEE.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. 2021. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 492–498. IEEE.

Heiga Zen and Haşim Sak. 2015. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4470–4474. IEEE.

Detecting ChatGPT-Generated Text with GZIP-KNN: A No-Training, Low-Resource Approach

Matthias Berchtold[*] and **Sandra Mitrović**[†] and **Davide Andreoletti**[‡]
and **Daniele Puccinelli**[‡] and **Omran Ayoub**[‡]
[*] University of Applied Sciences and Arts of Southern Switzerland (SUPSI), Switzerland
[†] Dalle Molle Institute for Artificial Intelligence (IDSIA), USI/SUPSI, Switzerland
[‡] Institute of Information Systems and Networking (ISIN), SUPSI, Switzerland

Abstract

Text classification is a fundamental Natural Language Processing task that is mostly addressed with resource-intensive transformer architectures. Researchers are continuously investigating lightweight alternatives without compromising predictive efficacy. A lightweight alternative called *Gzip-KNN* that combines the compression capability of Gzip with the K-nearest neighbors (KNN) classifier has been recently proposed. In this paper, we investigate the potential of Gzip-KNN for the detection of AI-generated text, notably ChatGPT-generated content. We compare its performance to several streamlined machine learning models such as Logistic Regression, eXtreme Gradient Boosting, and Gated Recurrent Unit. Our evaluation considers predictive accuracy, training duration, and inference speed, all while adjusting the available data in in- and out-of-domain contexts. Our experimental results highlight that *Gzip-KNN* achieves high predictive performance, often surpassing other models, especially when operating on a limited dataset for inference. Nonetheless, its extended inference time restricts its utility in time-sensitive scenarios. Intriguingly, *Gzip-KNN* manages to match the performance of other tested approaches even when utilizing a very limited percentage of the available data.

Keywords ChatGPT, Generative Language Models, Bots, GZIP

1 Introduction

The task of text classification, i.e., the categorization of a text into predefined classes, is fundamental in the domain of Natural Language Processing (NLP). The general approach involves designing a function that maps texts to their corresponding classes. Such function is generally obtained with supervised machine learning. Specifically, supervised training is performed by tuning the parameters of the models to minimize the error between estimated and ground truth classes. The complexity of the training process highly depends on the number of model parameters, which typically ranges from a few to millions.

Conventional approaches for text classification rely on complex models such as neural networks and in particular, transformers-based architectures, which are characterized by millions of parameters. These models yield remarkable predictive performance at the cost of a high training complexity. Indeed, training these models is expensive in terms of the amounts of data required, computational power, and training time (Chollet, 2017; Thompson et al., 2020). Therefore, employing such models for text classification may be an overkill. Rather than relying solely on large models, there is a growing interest in rediscovering lightweight approaches that can match the predictive accuracy of more complex models while requiring less computational power and training data (Fournier et al., 2023; Gururangan et al., 2019; Pan et al., 2019).

Recently, Jiang et al. (Jiang et al., 2023) proposed a lightweight methodology for text classification based on the combination of data compressing techniques (the *Gzip* compressor) and a low-complexity classifier (KNN, i.e., the K-nearest neighbors algorithm).

The proposed approach is referred to as *Gzip-KNN*, and is discussed in detail in Section 3. One of the distinctive traits of *Gzip-KNN* is the high computational efficiency due to its simple underlying components and non-parametric nature. In fact, the absence of tunable parameters drastically simplifies the training process. The intuition behind *Gzip-KNN* is that samples belonging to the same class are inherently more regular compared to samples from different classes. Hence, a lossless compression technique, such as the well-known *Gzip* algorithm, can be used to obtain representations that capture this intrinsic regularity. Subsequently, the representation of the sample undergoing classification is compared with the representations of the training samples using a novel distance metric. This process yields a distance matrix, which serves as the input for a k-nearest-neighbor classifier. In their study, the authors compare the predictive performance of *Gzip-KNN* with that of deep learning techniques and the Google Bidirectional Encoder Representations from Transformers (BERT) model. Experimental results show that *Gzip-KNN* is competitive with deep learning methods and can outperform BERT in out-of-domain benchmark datasets, exemplifying its robustness in handling unseen data distributions.

In our work, we extend on the previous study by focusing on the detection of AI-generated text. More specifically, we evaluate the potential of *Gzip-KNN* for the detection of texts generated by ChatGPT. We frame the problem as a supervised classification task, where the objective is to learn a mapping between a representation of the text and a binary variable, which is 1 if the text is generated by ChatGPT, and 0 otherwise. Then, we compare the performance of *Gzip-KNN*, in terms of predictive power, training time, inference time and memory footprint, to that of other approaches. In particular, we consider both lightweight models, such as logistic regression and eXtreme Gradient Boosting (XGB), and more complex approaches, namely the Gated Recurrent Unit (GRU). We refrain from considering pre-trained models as our aim is to compare *Gzip-KNN* to low-resource approaches. To systematically discuss our findings, we pose the following research questions (RQs):

RQ1) To what extent can *Gzip-KNN* detect ChatGPT-generated text? Can *Gzip-KNN* outperform traditional ML-supervised approaches in terms of predictive performance?

RQ2) How does *Gzip-KNN* compare to other approaches in terms of training time and inference time?

RQ3) Can *Gzip-KNN* outperform traditional ML approaches in an out-of-domain context? And in a data-constrained and inference-time-constrained scenarios?

To address these RQs, we conduct two experiments. In the first, we analyze the trade-off between predictive performance and complexity of *Gzip-KNN* and the supervised learning approaches in an in-domain context. In the second, we perform evaluations considering constraints on available data and on inference time in an out-of-domain context. The experimental findings demonstrate that *Gzip-KNN* exhibits strong predictive performance, surpassing alternative methods, even when making predictions with a limited amount of data. However, it does come with the drawback of increased inference time, which restricts its suitability to situations where rapid decision-making is not critical. Nevertheless, the results also indicate that *Gzip-KNN* can deliver comparable performance to other methods when utilizing only a small fraction of the available data in an out-of-domain context.

The paper is organized as follows. Section 2 discusses related work. Section 3 describes the *Gzip-KNN* approach proposed in (Jiang et al., 2023). In Section 4 we describe the datasets and experimental setup, and in Section 5 we present and discuss experiment results. Finally, Section 6 concludes the paper.

2 Related Work

The detection of AI-generated text is currently receiving a great deal of attention, as the proliferation of AI-generated text, particularly from advanced language models such as ChatGPT, has led to growing concerns about the authenticity and reliability of textual content across diverse domains (Guo et al., 2023; Khalil and Er, 2023; Tian and Cui, 2023). Moreover, as AI-generated content becomes more prevalent in online interactions, news articles, customer support chats, and creative writing, the need to accurately distinguish between human-generated and AI-generated text has gained paramount significance.

The community has dedicated substantial efforts to developing sophisticated machine learning models capable of detecting AI-generated content (Pegoraro et al., 2023; Liu et al., 2023; Guo et al., 2023; He et al., 2023). In particular, zero-shot and one-shot techniques have gained attention as innovative approaches for text classification in general and for identifying AI-generated text in particular (Mitchell et al., 2023; Liu et al., 2023; Yan et al., 2018).

Other approaches rely on statistical properties (Gehrmann et al., 2019), linguistic features (Ma et al., 2023; Guo et al., 2023), information-theorical metrics such as entropy (Gehrmann et al., 2019) and perpexity (Tian and Cui, 2023; Guo et al., 2023), topological features (of attention maps generated by the transformer model)

(Kushnareva et al., 2021), Transformers (Bleumink and Shikhule, 2023), pretrained language models without (Bakhtin et al., 2019) or with fine-tuning (Solaiman et al., 2019; Mitrović et al., 2023; Chakraborty et al., 2023; Ippolito et al., 2019; Guo et al., 2023; Chiang et al., 2023; Ma et al., 2023), where in particular GPT-2 Output Detector is frequently used (Gao et al., 2023; Anderson et al., 2023).

While these proposed methods may achieve the desired predictive performance on in- and out-of-domain data, their demanding computational requirements and memory footprint is a substantial obstacle to their deployment. *Gzip-KNN* presents a lightweight and resource-efficient alternative to complex solutions for AI-generated text detection, leveraging an innovative combination of approaches (e.g., compression techniques) to perform text classification without prior training.

3 Gzip and K-Nearest Neighbors for Text Classification

Algorithm 1 Text Classification using Gzip-KNN

```
Sample $t$ to be classified
Training dataset $\mathcal{D} = \{s_1, s_2, \ldots, s_n\}$
Number of nearest neighbors $k$
function CLASSIFY($t, \mathcal{D}, k$)
    Compress $t$ using Gzip (denote as $gzip(t)$)
    for each sample $s$ in $\mathcal{D}$ do
        Compress $s$ using Gzip ($gzip(s)$)
        Compute Normalized Compression Distance (NCD) between $gzip(s)$ and $gzip(t)$
        Store NCD in a distance list
    end for
    Find the indices of the $k$ smallest distances in the distance list
    Retrieve the corresponding $k$ nearest neighbors' classes
    Count the occurrences of each class among the $k$ neighbors
    Pick the majority class as the target label for $t$
end function
```

In this Section, we describe the various steps executed by the *Gzip-KNN* algorithm to classify a sample text *t*. The corresponding pseudocode is shown in Algorithm 1. The first step involves compressing *t* using the *Gzip* algorithm. Then, for each sample s in the training dataset, the text is similarly compressed using *Gzip*, and subsequently, the Normalized Compression Distance (NCD) between the compressed form of *s* and *t* is calculated (see Equation 1).

$$NCD(t, s) = \frac{C(st) - \min\{C(t), C(s)\}}{\max\{C(t), C(s)\}} \tag{1}$$

where st represents the concatenation of texts *t* and *s*, while $C(\cdot)$ is the length of a text compressed using *Gzip*.

The NCD serves as a measure that indicates the extent of information shared between two distinct texts. When two texts exhibit substantial shared content, their concatenation yields a more efficient compression outcome, resulting in a reduced NCD value. Therefore, since texts belonging to the same class typically share a greater degree of common attributes compared to texts from distinct classes, the NCD value can be leveraged in the task of text classification. Specifically, the *Gzip-KNN* algorithm uses the NCD distance computation as the basis for identifying the k-nearest neighbors of a reference text *t* within the training set. Finally, the target text *t* is classified based on the majority label among the selected k-nearest neighbors.

The absence of tunable parameters makes training lightweight and straightforward. However, it must be noted that classifying a text sample *t* requires repeating the concatenation between *t* and all the samples of the training set, which may result in a high inference time, especially with large datasets. In Section 5 we analyze the impact of the size of the training set on both predictive performance and inference time.

4 Dataset

We choose a labeled dataset consisting of human- (class 0) and ChatGPT-generated (class 1) responses to a set of queries. Specifically, ChatGPT answers were generated using GPT-3.5. These responses are provided in relation to a set of queries that encompass a wide range of open-ended questions. These questions were drawn from five diverse datasets, each contributing queries representative of a specific domain:

- *open_qa*: General queries on various topics sourced from the WikiQA dataset (Yang et al., 2015).

- *wiki_csai*: Queries related to specific concepts within the realm of information technology, gathered from Wikipedia (Guo et al., 2023).
- *finance*: Queries centered around finance-related subjects, obtained from the FiQA dataset (Maia et al., 2018).
- *medicine*: Queries focused on the field of medicine, collected from the Medical Dialog dataset (Zeng et al., 2020).
- *reddit_eli5*: Open-ended questions spanning various subjects, gathered from the ELI5 dataset (Fan et al., 2019).

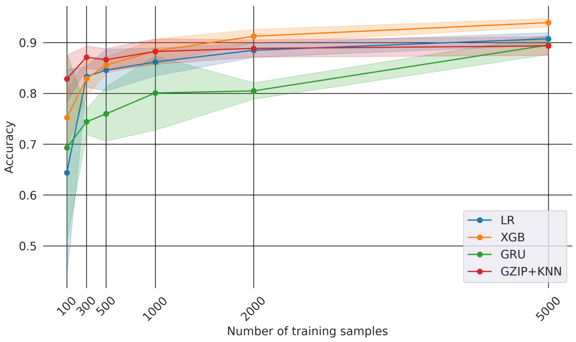
The human- and ChatGPT-generated responses are of similar length distribution (Guo et al., 2023).

5 Experimental Setup and Quantitative Evaluation

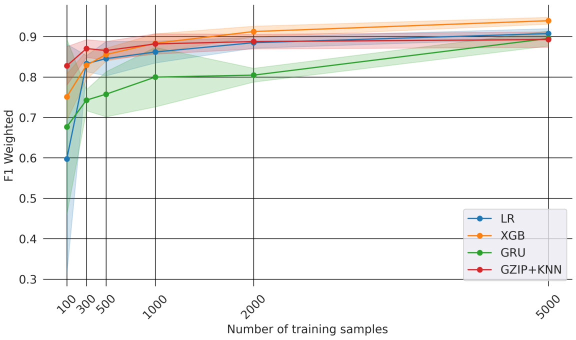
In this section, we present the results of our experiments, which aim to evaluate the *Gzip-KNN* algorithm for text classification along various dimensions. Specifically, in Section 5.1 we first evaluate the classification performance and the complexity of the approach, in terms of training time and inference time. Then, in Section 5.2, we assess the classification performance in an out-of-domain setting, where the algorithm is tested on datasets never seen during training.

5.1 Experiment 1: Performance vs. Complexity

Experimental Setup: We conduct this experiment while systematically varying the number of responses used during the training phase, all while ensuring a balanced distribution between the human-generated and ChatGPT-generated responses. Specifically, we consider a range of responses *n = 100, 300, 500, 1000, 2000, 5000*, which are randomly selected from the dataset. We adopt a 5-fold cross-validation methodology for each value of *n*. During the testing phase, we evaluate the approach using a set of 10,000 responses, selected randomly from the dataset and equally split between the two labels. We also ensure the same testing sets are used to evaluate the different models for fold. The aim of this analysis is to comprehensively assess the model's performance across different training data volumes while maintaining a consistent test set.



(a) Classification Accuracy for varying training size



(b) F1-score for varying training size

Figure 1: Classification performance of the LR, XGB, GRU and Gzip-KNN models, for varying sizes of the training set

We compare *Gzip-KNN* to Logistic Regression (LR), eXtreme Gradient Boosting (XGB), and Gated Recurrent Unit (GRU) along two main dimensions, namely predictive performance and model complexity. To evaluate the former, we consider traditional classification metrics, such as *accuracy* and *F1-score*. To evaluate the latter, we consider the training time $t_{training}$ and the inference time $t_{inference}$.

Predictive Performance: Figure 1(a) shows the mean accuracy and standard deviation of the four models with respect to the number of training samples, ranging between 100 and 5000 training samples. When only 100 training samples are used, *Gzip-KNN* achieves an accuracy of 0.83, outperforming XGB (0.75), GRU (0.69), and LR (0.65). The significant gap in accuracy reveals *Gzip-KNN*'s capability in text classification, and particularly, in detecting ChatGPT-generated text, with very little training data. The accuracy of all models shows a general upward trend in performance as the training dataset size increases with *Gzip-KNN* outperforming the other approaches. However, it is worth noticing that the performance gap between *Gzip-*

KNN and the other approaches narrows as the training dataset size grows, up to a number of training samples equal to 1000. Specifically, for a number of training samples of 1000, *Gzip-KNN* shows an accuracy of 0.88, just slightly higher than that of XGB (0.87), LR (0.86), and GRU (0.81). The accuracy achieved by all models continues to increase as the training set size becomes larger, except for *Gzip-KNN*, which seems to saturate around an accuracy of 0.89, outperformed by XGB, LR, and GRU, which achieve an accuracy of 0.94, 0.91 and 0.9 using 5000 training samples, respectively.

Figure 1(b) shows the weighted F1-score and its standard deviation achieved by the different approaches with respect to the number of training samples, ranging between 100 and 5000. Results in terms of F1-score show a similar trend to that of accuracy. For a number of training samples less than 1000, *Gzip-KNN* outperforms other approaches, achieving an F1-score of around 0.88. This confirms *Gzip-KNN*'s intrinsic ability to distill and compress information effectively, even when the dataset is not exceedingly large, and that its architecture inherently adapts to the complexity of the data, discerning relevant features and connections without the need for a large number of examples. For a higher number of training samples, the F1-score of *Gzip-KNN* saturates around 0.89 while that of other approaches continues to show a slight increase as the size of the training dataset increases.

Overall, results show that the performance differences between *Gzip-KNN* and the other models, namely, XGB, LR, and GRU, tend to diminish as the training size increases, which could be due to more data being available for XGB, LR, and GRU, reducing overfitting and improving generalization, while *Gzip-KNN* does not further benefit from more training samples. In other words, the performance gains achieved by the *Gzip-KNN* seem to saturate beyond a certain point of dataset size. Unlike traditional methods that tend to improve as more data is fed into their training pipelines, *Gzip-KNN* appears to capitalize on a specific threshold of data sufficiency. This suggests that, for *Gzip-KNN*, the emphasis should be placed not solely on increasing the dataset size, but rather on refining the compression and distance calculation mechanisms. Further research could delve into optimizing the interplay between these two components to extract more nuanced information and potentially push the *Gzip-KNN*'s performance boundaries.

Model Complexity: We first examine the complexity of the considered approaches in terms of training time and inference time. Table 1 shows the training time (in seconds), averaged over 10 different evaluations, for the various models with respect to the number of training observations. Results show that the training times of the various models exhibit distinct trends as the number of training observations increases. The *Gzip-KNN*, which demonstrates exceptional efficiency, consistently yielding remarkably low training times across, shows a slightly increasing trend, ranging between 0.005 seconds for 100 training observations to 0.093 seconds for 2000 training observations. LR and XGB demonstrate linear increments in training time with the expansion of training data, reaching up to 0.274 seconds for LR and 0.810 seconds for XGB, at 2000 training observations. GRU, on the contrary, shows a nuanced pattern, with training times displaying some fluctuations without a clear trend, ranging between 8.7 and 13.3 seconds, on average. These results highlight the clear advantage *Gzip-KNN* has over other models in terms of training time, which suggests its potential utility for scenarios demanding swift model deployment.

We now focus on the inference time. Table 2 reports the variation in inference times for different models, across various sizes of the training set. Notably, the *Gzip-KNN* approach consistently exhibited relatively higher inference times compared to the other models for all sizes of the training set, ranging between 5.9 seconds (when 100 observations are used to perform the inference) to 115.6 seconds (when 2000 observations are used to perform the inference). On the contrary, the inference time for other approaches is significantly lower (i.e., fluctuating around 1 second), and not dependent on the size of the training set. This shows that the *Gzip-KNN* approach introduces an additional significant computational overhead with respect to other approaches, particularly when the number of observations used for inference is relatively large.

Gzip-KNN: Performance vs. Complexity. Focusing our attention on *Gzip-KNN* approach, a distinct trade-off emerges between predictive power and the time required for inference. As illustrated in Figure 1, *Gzip-KNN* achieves a relatively high predictive performance (0.82 of accuracy and 0.85 of F1-score) even with a small number of obser-

Table 1: Training Time Results

Num. Training Observations	Training Time (seconds)			
	GRU	GZIP+KNN	LR	XGB
100	10.101	0.005	0.017	0.093
300	9.497	0.014	0.040	0.168
500	8.701	0.023	0.054	0.230
1000	14.179	0.048	0.111	0.458
2000	13.346	0.093	0.275	0.811
5000	28.238	0.225	0.496	1.697

Table 2: Inference Time Results

Num. Training Observations	Inference Time (seconds)			
	GRU	GZIP+KNN	LR	XGB
100	1.033	5.944	0.072	0.096
300	0.922	17.332	0.074	0.095
500	0.781	28.848	0.076	0.095
1000	1.030	58.784	0.091	0.100
2000	0.884	115.589	0.083	0.099
5000	0.927	286.129	0.077	0.103

vations employed for inference (100 observations). This, however, corresponds to a relatively elevated inference time of 5 seconds (see Table 2). The predictive performance of *Gzip-KNN* can be further improved to reach 0.9 accuracy by employing a larger set of observations for inference (1000 samples). However, this incurs a substantial increase in inference time, culminating in an extended duration of 58.5 seconds. This suggests that the *Gzip-KNN* approach can have a robust predictive performance, particularly when dealing with a constrained dataset. Yet, its value is limited to scenarios where prediction accuracy takes precedence over rapid inference. In conclusion, while *Gzip-KNN* offers a powerful tool for predictive tasks, its optimal use hinges on aligning its strengths with the specific requirements of the given application context.

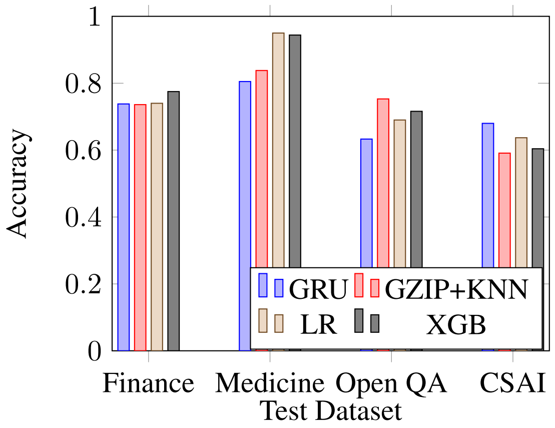
5.2 Experiment 2: Performance in Out-of-Domain Context

Experimental Setup: We now shift our attention to assessing the performance of the different methods in an out-of-domain context in different circumstances. Specifically, we perform two evaluations. In the first evaluation, we consider that a limited amount of data is available for training. The objective of this experiment is to quantify the capability of *Gzip-KNN* in detecting ChatGPT-generated text in an out-of-domain context and under the limitations of available data. We perform the training considering a part of the datasets, set at 1000 text samples, extracted from three specific contexts (e.g., from technology, finance and open QA datasets) with equal contribution and then perform the testing on a different dataset that corresponds to a different context (e.g., medicine). Note that while no formal training takes place for the *Gzip-KNN* approach, the inference still relies on the utilization of text samples, which are the training samples used to train the other ML models.

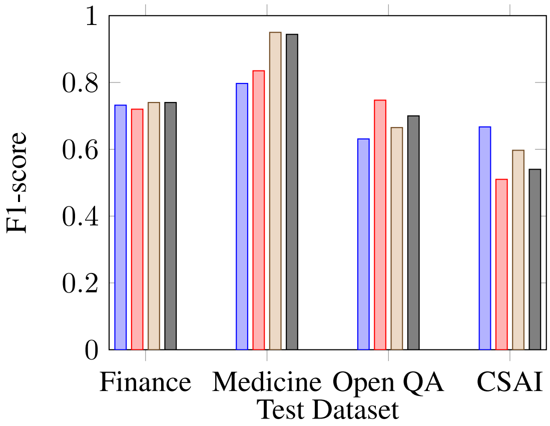
In the second evaluation, we introduce a constraint on inference time. To comply with the imposed inference time constraint, the size of the data used by *Gzip-KNN* at inference time must be restricted. Specifically, 600 samples are randomly taken from the training set, and used to perform the inference. On the contrary, for the other approaches, since using all data available for training does not heavily impact the inference time, we consider that all data available can be used for training. The objective of this experiment is to asses whether *Gzip-KNN* can outperform other models in an out-of-domain context even when a limit is imposed on inference time (and, therefore, on the amount of data that are required by *Gzip-KNN* to perform a text classification).

Out-of-Domain ChatGPT-generated Text Detection with Limited Data: Figures 2(a) and 2(b) show the accuracy and F1-score metrics, respectively, that are achieved by the four models when tested on the considered datasets. In general, XGB and LR tend to outperform other approaches, consistently achieving some of the highest performance levels across most datasets. For instance, in the Finance dataset, XGB achieves the best accuracy and F1-score (0.807 and 0.805, respectively) and ranks second in the Medicine and OpenQA datasets (e.g., its accuracy is 0.944 and 0.716, respectively). LR achieves the highest accuracy and F1-score in the Medicine dataset (0.95 for both metrics) and it ranks second in the Finance and CSAI datasets, with accuracy values of 0.74 and 0.69, respectively. *Gzip-KNN* generally achieves lower performance compared to alternative methods. However, it is important to note that *Gzip-KNN* reaches the best performance in the OpenQA dataset, surpassing alternative methods in both accuracy and F1-score, with values of 0.753 and 0.665, respectively. Additionally, in the Finance dataset, the performance of *Gzip-KNN* is only slightly lower than that of the alternatives (indeed, GRU, *Gzip-*

KNN, and LR all achieve an accuracy of around 0.74).



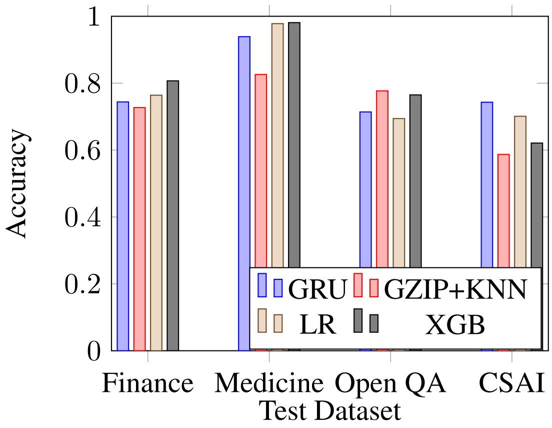
(a) Accuracy of Different Models on Various Datasets



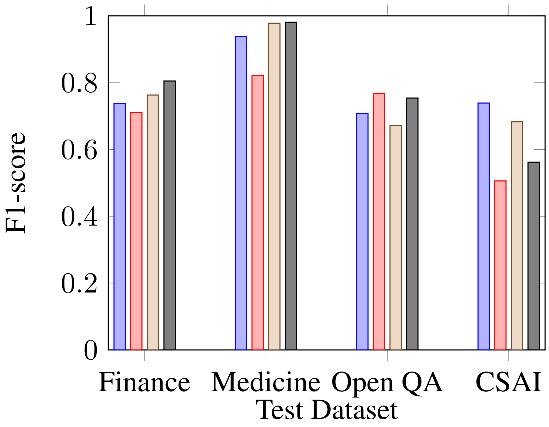
(b) F1-score of Different Models on Various Datasets

Figure 2: Comparison of accuracy and F1-score of different models for out-of-domain ChatGPT-generated text detection considering different test datasets.

Out-of-Domain ChatGPT-generated Text Detection under Constraint on Inference Time: Figures 3(a) and 3(b) show the accuracy and F1-score metrics of the four different approaches for each of the cases, respectively. Notably, *XGB* exhibits consistently high accuracy and F1-scores, specifically for Medicine and Finance (0.981 accuracy in both cases), outperforming other approaches in all cases except for when testing over the CSAI dataset. GRU also shows a similar performance outperforming other approaches for the case of testing over the CSAI dataset. *LR* demonstrates competitive performance outperforming GRU in some cases while *Gzip-KNN* achieves the highest accuracy on the Medicine dataset (0.826) and the lowest on the CSAI dataset (0.587). With respect to other approaches, *Gzip-KNN* achieved generally lower, yet comparable, accuracy and F1-scores, except when testing on Open QA, where it achieved the highest accuracy and F1-score. Note that *Gzip-KNN* uses only 600 text samples for this experiment, while other approaches utilize all available datasets. This shows that *Gzip-KNN* can achieve performance in out-of-domain ChatGPT-generated text detection when using a significantly small amount of data (a portion of the dataset) comparable to that of other approaches (in this case, LR, GRU, and XGB) when trained on the entire dataset.



(a) Accuracy of Different Models on Various Datasets



(b) F1-score of Different Models on Various Datasets

Figure 3: Comparison of accuracy and F1-score of different models for out-of-domain ChatGPT-generated text detection considering different test datasets.

6 Conclusion

In this work, we evaluate the effectiveness of a recently proposed algorithm, *Gzip-KNN*, in the task of detection of ChatGPT-generated text. The *Gzip-KNN* algorithm combines compression techniques with the k-nearest neighbors (KNN) algorithm for classification, resulting in a lightweight solution compared to traditional techniques used for text classification. Specifically, we compare this approach with LR, XGB, and GRU, in terms of classification performance and model complexity in various scenarios. Obtained results show that the *Gzip-KNN* algorithm outperforms the alternatives in terms of classification performance in situations where the training dataset is limited in the number of samples. However, such an advantage comes also with an increased inference time, which is significantly higher for *Gzip-KNN* than for the other approaches. Finally, we also evaluated the classification performance of the approaches in an out-of-domain setting, where the models are tested on a set never seen during training. These experiments have shown that *Gzip-KNN* can yield comparable classification performance to the other methods while only utilizing a significantly lower amount of training data.

References

Nash Anderson, Daniel L Belavy, Stephen M Perle, Sharief Hendricks, Luiz Hespanhol, Evert Verhagen, and Aamir R Memon. 2023. Ai did not write this manuscript, or did it? can we trick the ai text detector into generated texts? the potential future of chatgpt and ai in sports & exercise medicine manuscript generation.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.

Arend Groot Bleumink and Aaron Shikhule. 2023. Keeping ai honest in education: Identifying gpt-generated text. *Edukado AI Research*, pages 1–5.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv preprint arXiv:2304.04736*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Francois Chollet. 2017. The limitations of deep learning. *Deep learning with Python*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. 2023. A practical survey on faster and lighter transformers. *ACM Computing Surveys*, 55(14s):1–40.

Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. 2023. Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers. *NPJ Digital Medicine*, 6(1):75.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A Smith. 2019. Variational pretraining for semi-supervised text classification. *arXiv preprint arXiv:1906.02242*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828.

Mohammad Khalil and Erkan Er. 2023. Will chatgpt get you caught? rethinking of plagiarism detection. *arXiv preprint arXiv:2302.04335*.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. *arXiv preprint arXiv:2109.04825*.

Zhengliang Liu, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Wei Liu, Dinggang Shen, Quanzheng Li, et al. 2023. Deid-gpt: Zero-shot medical text de-identification by gpt-4. *arXiv preprint arXiv:2303.11032*.

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. Ai vs. human – differentiation analysis of scientific content generation.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

Chongyu Pan, Jian Huang, Jianxing Gong, and Xingsheng Yuan. 2019. Few-shot transfer learning for text classification with lightweight word embedding based models. *IEEE Access*, 7:53296–53304.

Alessandro Pegoraro, Kavita Kumari, Hossein Fereidooni, and Ahmad-Reza Sadeghi. 2023. To chatgpt, or not to chatgpt: That is the question! *arXiv preprint arXiv:2304.01487*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. 2020. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Leiming Yan, Yuhui Zheng, and Jie Cao. 2018. Few-shot learning for short text classification. *Multimedia Tools and Applications*, 77:29799–29810.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. 2020. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online. Association for Computational Linguistics.

Data Bias According to Bipol: Men are Naturally Right and It is the Role of Women to Follow Their Lead

Irene Pagliai[*1], **Goya van Boven**[*2], **Tosin Adewumi**[*†], **Lama Alkhaled**[†], **Namrata Gurung**[3], **Isabella Södergren**[4] and **Elisa Barney**[†]
[1]University of Göttingen, Germany, [2]Utrecht University, the Netherlands, [*†]Machine Learning Group, LTU, Sweden, [3]QualityMinds GmbH, Germany, [4]Digital Services and Systems, LTU.
[1]irene.pagliai@uni-goettingen.de, [2]j.g.vanboven@students.uu.nl, [†]firstname.lastname@ltu.se,
[3]namrata.gurung@qualityminds.de [4]isasde-5@student.ltu.se | [*]Joint first authors

Abstract

We introduce new large labeled datasets on bias in 3 languages and show in experiments that bias exists in all 10 datasets of 5 languages evaluated, including benchmark datasets on the English GLUE/SuperGLUE leaderboards. The 3 new languages give a total of almost 6 million labeled samples and we benchmark on these datasets using SotA multilingual pretrained models: mT5 and mBERT. The challenge of social bias, based on prejudice, is ubiquitous, as recent events with AI and large language models (LLMs) have shown. Motivated by this challenge, we set out to estimate bias in multiple datasets. We compare some recent bias metrics and use bipol, which has explainability in the metric. We also confirm the unverified assumption that bias exists in toxic comments by randomly sampling 200 samples from a toxic dataset population using the confidence level of 95% and error margin of 7%. Thirty gold samples were randomly distributed in the 200 samples to secure the quality of the annotation. Our findings confirm that many of the datasets have male bias (prejudice against women), besides other types of bias. We publicly release our new datasets, lexica, models, and codes.

1 Introduction

The problem of social bias in data is a pressing one. Recent news about social bias of artificial intelligence (AI) systems, such as Alexa[1] and ChatGPT,[2] shows that the age-old problem persists with data, which is used to train machine learning (ML) models. Social bias is the inclination or prejudice for, or against, a person, group or idea, especially in a way that is considered to be unfair, which may be based on race, religion or other factors (Bellamy et al., 2018; Antoniak and Mimno, 2021; Mehrabi et al., 2021; Alkhaled et al., 2023). It can also involve stereotypes that generalize behavior to groups (Brownstein, 2019). It can unfairly skew the output of ML models (Klare et al., 2012; Raji et al., 2020). Languages with fewer resources than English are also affected (Rescigno et al., 2020; Chávez Mulsa and Spanakis, 2020; Kurpicz-Briki, 2020). For example, in Italian, the female gender is under-represented due to the phenomena such as the “inclusive masculine" (when the masculine is over-extended to denote groups of both male and female referents) (Luccioli et al.; Vanmassenhove and Monti, 2021).

In this work, we are motivated to address the research question of *how much bias exists in the text data of multiple languages, if at all bias exists in them*? We particularly investigate 6 benchmark datasets on the English GLUE/SuperGLUE leaderboards (Wang et al., 2018, 2019) and one dataset each for the other 4 languages: Italian, Dutch, German, and Swedish. First, we train SotA multilingual Text-to-Text Transfer Transformer (mT5) (Xue et al., 2021) and multilingual Bidirectional Encoder Representations from Transformers (mBERT) models for bias classification on the multi-axes bias dataset (MAB) for each language, in a similar setup as Alkhaled et al. (2023). For the evaluations, we search through the literature to compare different metrics or evaluation methods as shown in Table 1 and discussed in Section 2. This motivates our choice of bipol, the multi-axes bias metric, which we then compare in experiments with a lexica baseline method. In addition, to confirm the unverified assumption that toxic comments contain bias (Sap et al., 2020; Alkhaled et al., 2023), we annotate 200 randomly-selected samples from the training set of the English MAB.

Our Contributions

- We make available new large labeled datasets on bias of almost 2 million samples each for

---

[1] bbc.com/news/technology-66508514

[2] bloomberg.com/news/newsletters/2022-12-08/chatgpt-open-ai-s-chatbot-is-spitting-out-biased-sexist-results

Metric/Evaluator	Axis	Terms
Winogender (Rudinger et al., 2018)	1	60
WinoBias (Zhao et al., 2018)	1	40
StereoSet (Nadeem et al., 2021)	4	321
GenBiT (Sengupta et al., 2021)	1	-
CrowS-Pairs (Nangia et al., 2020)	9	3,016
Bipol (Alkhaled et al., 2023)	>2, <13	>45, <466

Table 1: Comparison of some bias evaluation methods.

3 languages: Italian, Dutch, and German.[3]

- We make available lexica of sensitive terms for bias detection in the 3 languages.
- We confirm the unverified assumption in the underlying datasets of MAB (Social Bias Inference Corpus v2 (SBICv2) and Jigsaw) (Alkhaled et al., 2023) that toxic comments contain bias.

The rest of this paper is organized as follows. In Section 2, we discuss the literature review of related work. In Section 3, we briefly discuss the *bipol* metric. In Section 4, we explain the steps involved in the methodology and the datasets we use. In Section 5, we present our findings and discuss them. In Section 6, we end with the conclusion and possible future work.

2 Literature Review

Although English usually gets more support and attention in the literature, there have been attempts at measuring and mitigating bias in other languages. Testing for the presence of bias in Italian often has a contrastive perspective with English, with a focus on gender bias (Gaido et al., 2021; Rescigno et al., 2020). MuST-SHE (Bentivogli et al., 2020) and gENder-IT (Vanmassenhove and Monti, 2021) are examples of gender bias evaluation sets. Going beyond gender bias, Kurpicz-Briki and Leoni (2021) and Huang et al. (2020) also identified biases related to people's origin and speakers' age. It is essential to remember that the mentioned biases can be vehicles for misogynous and hateful discourse (El Abassi and Nisioi, 2020; Attanasio et al., 2022; Merenda et al., 2018).

Bias studies for Dutch mostly consider binary gender bias. Chávez Mulsa and Spanakis (2020) investigate gender bias in Dutch static and contextualized word embeddings by creating Dutch versions of the Word/Sentence Embedding Association Test (WEAT/SEAT) (Caliskan et al., 2017; May et al., 2019). WEAT measures bias in word embeddings and can be limited in scope, in addition to having sensitivity to seed words. McCurdy and Serbetci (2020) perform a similar evaluation in a multilingual setup to compare the effect of grammatical gender saliency across languages. Several works use different NLP techniques to evaluate bias in corpora of Dutch news articles (Wevers, 2019; Kroon et al., 2020; Kroon and van der Meer, 2021; Fokkens et al., 2018) and literary texts (Koolen and van Cranenburgh, 2017).

In Kurpicz-Briki (2020), bias is measured with regards to place of origin and gender in German word embeddings using WEAT (Caliskan et al., 2017). In Kurpicz-Briki and Leoni (2021), an automatic bias detection method (BiasWords) is presented, through which new biased word sets can be identified by exploring the vector space around the well-known word sets that show bias. In the template-based study of Cho et al. (2021), on gender bias in translations, the accuracy of gender inference was measured for multiple languages including German. It was found that, particularly for German, the inference accuracy and disparate impact were lower for female than male, implying that certain translations were wrongly performed for cases that required female inference. Since German is a grammatically gendered, morphologically rich language, Gonen and Goldberg (2019) found that debiasing methods of Bolukbasi et al. (2016) were ineffective on german word embeddings.

For Swedish, the main focus of bias research appears to be on gender. Sahlgren and Olsson (2019) show with their experiments that gender bias is present in pretrained Swedish language models. Katsarou et al. (2022) and Precenth (2019) found that the male gender tends to be associated with higher-status professions. A study with data from mainstream news corpora by Devinney et al. (2020) shows that women are associated with concepts like family, communication and relationships.

3 Bipol

For the purpose of this work, we summarize *bipol* here but details are discussed in Alkhaled et al. (2023). The *bipol* metric uses a two-step mechanism for estimating bias in text data: binary classification and sensitive term evaluation using lexica. It has maximum and minimum values of 1 and 0, respectively. Bipol is expressed in Equations 1b and 1c from the main Equation 1a, where b_c is

---

[3] github.com/LTU-Machine-Learning/bipolmulti

the classification component and b_s is the sensitive term evaluation component.

$$b = \begin{cases} b_c.b_s, & \text{if } b_s > 0 \\ b_c, & \text{otherwise} \end{cases} \tag{1a}$$

$$b_c = \frac{tp + fp}{tp + fp + tn + fn} \tag{1b}$$

$$b_s = \frac{1}{r}\sum_{t=1}^{r}\left(\frac{1}{q}\sum_{x=1}^{q}\left(\frac{|\sum_{s=1}^{n} a_s - \sum_{s=1}^{m} c_s|}{\sum_{s=1}^{p} d_s}\right)_x\right)_t \tag{1c}$$

In step 1, a trained model is used to classify all the samples. The ratio of the biased samples to the total samples predicted is determined. The ***tp***, ***fp***, ***tn***, and ***fn*** are values of the true positives, false positives, true negatives, and false negatives, respectively. Since there's hardly a perfect classifier, the positive error rate is usually reported. False positives are known to exist in similar classification systems like spam detection and automatic hate speech detection (Heron, 2009; Feng et al., 2018).

Step 2 is similar to *term frequency-inverse document frequency* (TF-IDF) in that it is based on term frequency (Salton and Buckley, 1988; Ramos et al., 2003), Biased samples from step 1 are evaluated token-wise along all possible bias axes, using all the lexica of sensitive terms. An axis is a domain such as gender or race. Tables 2 and 3 provide the lexica sizes. For English and Swedish, we use the same lexica released by Alkhaled et al. (2023) and Adewumi et al. (2023b), respectively. For the other 3 languages, we create new lexica of terms (e.g. she & her) associated with specific gender or stereotypes from public sources.[4] Some of the terms in the lexica were selected from the sources based on the topmost available. These may also be expanded as needed, since bias terms are known to evolve (Haemmerlie and Montgomery, 1991; Antoniak and Mimno, 2021). The non-English lexica are small because fewer terms are usually available in other languages compared to the high-resource English language and we use the same size across the languages to be able to compare performance somewhat. The Appendix lists these terms.

Equation 1c first finds the absolute difference between the two maximum summed frequencies in the types of an axis ($|\sum_{s=1}^{n} a_s - \sum_{s=1}^{m} c_s|$), where n and m are the total terms in a sentence along an axis. For example, in the sentence *´Women!!! PERSON taught you better than that. Shame on you!’*, female terms = 1 while male terms = 0. This is then divided by the summed frequencies of all the terms (d_s) in that axis ($\sum_{s=1}^{p} d_s$). The operation is performed for all axes (q) and the average taken ($\frac{1}{q}\sum_{x=1}^{q}$). It is performed for all the biased samples (r) and the average taken ($\frac{1}{r}\sum_{t=1}^{r}$).

Axis	Type 1	Type 2	Type 3
Racial	84 (black)	127 (white)	
Gender	76 (female)	46 (male)	
Religious	180 (christian)	465 (muslim)	179 (hindu)

Table 2: English lexica sizes (Alkhaled et al., 2023). An example of racial-black is *negro*.

Axis	Type 1	Type 2
Gender	17 (female)	17 (male)
Racial	10 (black)	10 (white)

Table 3: Italian, Dutch, German and Swedish Lexica sizes. An example of gender-female is *woman*.

4 Methodology

4.1 Datasets

4.1.1 MAB

The Italian, Dutch and German datasets were machine-translated from MAB[5] with the high-quality Helsinki-NLP model (Tiedemann and Thottingal, 2020). Each translation took about 48 hours on one GPU. Examples from the data are provided in Table 4. Table 5 provides statistics about the datasets. For quality control (QC), we verified translation by back-translating some random samples using Google NMT. Personal identifiable information (PII) was removed from the MAB dataset using the spaCy library. The 3 datasets are used to train new bias classifiers. We also train on the original English and the Swedish.

Machine-Translation issues: Culture-specific biases may not be represented in the MAB versions for the translated languages because the original dataset is in English. This is a limitation. However, bias is also a universal concern, such that there are examples that span across cultures. For instance, the examples in Table 4 are of universal concern because individuals with non-conforming bodies and women should be respected, regardless of culture or nationality. Hence, the MAB versions

---

[4] fluentu.com/blog/italian/italian-nouns, en.wiktionary.org/wiki/Category:Italian_offensive_terms, Dutch_profanity, Category:German_ethnic_slurs

[5] The reference provides details of the annotation of the base data.

comment_text	**label**	**old_id**	**id**
A fitness ranking that drops each month as more and more fat people move to Colorado.	biased	6231681	167
Een fitness ranking die elke maand daalt als meer en meer dikke mensen verhuizen naar Colorado.			
Ein Fitness-Ranking, das jeden Monat fällt, wie mehr und mehr fette Menschen nach Colorado zu bewegen.			
Un ranking di fitness che scende ogni mese come sempre più persone grasse si spostano in Colorado.			
En fitness ranking som sjunker varje månad som fler och fler feta människor flyttar till Colorado.			
Women ! ! ! PERSON taught you better than that. Shame on you!	biased	6233025	628
Vrouwen ! ! ! PERSOON heeft je beter geleerd dan dat. Je moet je schamen!			
Frauen!!!! PERSON lehrte Sie besser als das. Schande über Sie!			
Donne ! ! ! Person ti ha insegnato meglio di così, vergognati!			
Kvinnor ! ! !- Han lärde dig bättre än så. Skäms på dig!			

Table 4: **English**, **Dutch**, **German**, **Italian**, and **Swedish** examples from the MAB dataset. "PERSON" is the anonymization of a piece of personal identifiable information (PII) in the dataset.

Set	**Biased**	**Unbiased**	**Total**
Training	533,544	1,209,433	1,742,977
Validation	32,338	69,649	101,987
Test	33,470	68,541	102,011
	599,352	1,347,623	1,946,975

Table 5: MAB dataset split

are relevant for bias detection, though they were translated.

4.1.2 Evaluation datasets

Ten datasets are evaluated for bias in this work. All are automatically preprocessed before evaluation, the same way the training data were preprocessed. This includes removal of IP addresses, emojis, URLs, special characters, emails, extra spaces, numbers, empty text rows, and duplicate rows. All texts are then lowercased.

We selected datasets that are available on the HuggingFace (Wolf et al., 2020) Datasets. We evaluated the first 1,000 samples of each training split due to resource constraints. The understanding is that if bias is detected in these samples, then scaling over the entire dataset means there's probability of more bias. For English, we evaluated the sentence column of Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019), the sentence column of Question-Answering Natural Language Inference (QNLI) (Wang et al., 2018), the sentennce1 column of Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), the premise column of Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018), the premise column of the CommitmentBank (CB) dataset (De Marneffe et al., 2019), and the passage column of Reading Comprehension with Commonsense Reasoning Dataset (ReCoRD) (Zhang et al., 2018). For Italian, we evaluated the context column of the Stanford Question Answering Dataset (SQuAD) (Croce et al., 2018; Rajpurkar et al., 2016); for Dutch, the sentence1 column of the Semantic Textual Similarity Benchmark (STSB) (Cer et al., 2017); for German, the text column of the German News Articles Datasets 10k (GNAD10) (Schabus et al., 2017); for Swedish, the premise of the CB.

4.2 Annotation for the assumption confirmation

To verify the assumption that toxic comments contain bias, we randomly selected 200 samples from the training set of MAB-English for annotation on Slack, an online platform. The selection of 200 samples is based on an error margin of 7% and a confidence level of 95%. To ensure high-quality annotation, we use established techniques for this task: 1) the use of gold (30) samples, 2) multiple (i.e. 3) annotators, and 3) minimum qualification of undergraduate study for annotators. Each annotator was paid 25 U.S. dollars and the it took about 2 hours to complete the annotation on average. We mixed the 30 gold samples with the 200, to verify the annotation quality of each annotator, as they were required to get, at least, 16 correctly for their annotation to be accepted. The 30 gold samples are samples with unanimous agreement in the original Jigsaw or SBICv2 data, which make up the MAB. We provide inter-annotator agreement (IAA) using Jaccard similarity coefficient (intersection over union) and credibility unanimous score (CUS) (Adewumi et al., 2023a) (intersection over sample size).

4.3 Experiments

We selected two state-of-the-art (SotA) pre-trained, multilingual models for experiments to compare their macro F1 performance: mT5-small and mBERT-base. These are from the HuggingFace hub. We further report the mT5 positive error rate of predictions. The mT5-small has 300 million parameters (Xue et al., 2021) while mBERT-Base has 110 million parameters. We trained only on the MAB datasets and evaluated using only the mT5 model, the better model of the 2, as will be observed in Section 5. For the CB and ReCoRD datasets, we evaluate all samples since they contain only about 250 and 620 entries, respectively. We used wandb (Biewald, 2020) for hyper-parameter exploration, based on Bayesian optimization. For mT5, we set the maximum and minimum learning rates as 5e-5 and 2e-5 while the maximum and minimum epochs are 20 and 4, respectively. One epoch is equivalent to the ratio of the total number of samples to the batch size (i.e. the steps). We used a batch size of 8 because higher numbers easily resulted in memory challenges.

For mBERT, we set the learning rates and epochs as with mT5. However, we explore over batch sizes of 8, 16 and 32. For both models, we set the maximum input sequence length to 512. Training took, on average, about 7.3 hours per language per epoch for mBERT while it was 6 hours for mT5. For all the experiments, we limit the run counts to 2 per language because of the long training time each takes on average. The average scores of the results are reported. The saved models with the lowest losses were used to evaluate the datasets. All the experiments were performed on two shared Nvidia DGX-1 machines that run Ubuntu 20.04 and 18.04. One machine has 8 x 40GB A100 GPUs while the other has 8 x 32GB V100 GPUs.

The lexica baseline, compared in experiments, is similar to the equation of the second step in bipol. It does not consider bias semantically and uses term frequencies, similarly to TF-IDF. It uses the same lexica as bipol. Its maximum and minimum values are 1 and 0, respectively.

5 Results and Discussion

From Table 6, we observe that all mT5 results are better than those of mBERT across the languages. The two-sample t-test of the difference of means between all the corresponding mT5 and mBERT scores have p values < 0.0001 for alpha of 0.05, showing the results are statistically significant. It appears better hyper-parameter search may be required for the mBERT model to converge and achieve better performance. The best macro F1 result is for English mT5 at 0.787. This is not surprising, as English has the largest amount of training data for the pre-trained mT5 model (Xue et al., 2021). This occurred at the learning rate of 2.9e-5 and step 1,068,041.

	macro F1 $\uparrow$ (s.d.)		**mT5 error** $\downarrow$
MAB version	**mBERT**	**mT5**	**fp/(fp+tp)**
English	0.418 (0.01)	0.787 (0)	0.261
Italian	0.429 (0)	0.768 (0)	0.283
Dutch	0.419 (0.01)	0.768 (0)	0.269
German	0.418 (0.01)	0.769 (0)	0.261
Swedish	0.418 (0.01)	0.768 (0)	0.274

Table 6: Average F1 scores on the validation sets.

Figures 1 and 2 depict the validation sets macro F1 and loss line graphs for the 2 runs for the 5 languages, respectively. From Table 7, we observe that all the evaluated datasets have biases, though seemingly little (but important) when compared to the maximum of 1. We say important because many of the datasets contain small number of sam-

	bipol scores		↓ (s.d.)	
English	b_c	b_s	**bipol *(b)***	**baseline ↓**
CB	0.096	0.875	0.084 (0)	0.88
CoLA	0.101	0.943	0.095 (0)	0.958
ReCoRD	0.094	0.852	0.025 (0)	0.829
MRPC	0.048	0.944	0.045 (0)	0.957
MNLI	0.063	0.833	0.053 (0)	0.965
QNLI	0.03	0.933	0.028 (0)	0.945
Italian				
SQuAD	0.014	0	0.014 (0)	0.989
Dutch				
STSB	0.435	0.992	0.432 (0)	0.987
German				
GNAD10	0.049	0.502	0.025 (0)	1
Swedish				
CB	0.08	0.938	0.075 (0)	0.97

Table 7: Average bipol & lexica baseline scores.

ples yet they can be detected. Furthermore, a low value does not necessarily diminish the weight of the effect of bias in society or the data but we leave the discussion about what amount should be tolerated open for the NLP community. Our recommendation is to have a bias score as close to zero as possible. On the other hand, the lexica baseline appears overly confident of much more bias, which is incorrect because the method fails to exclude unbiased text in its evaluation, which is a shortcoming of methods based solely on it. The Dutch STSB is higher than the other bipol scores because of the higher bipol classifier component score of 0.435, which may be because of the nature of the dataset.

5.1 Error analysis & qualitative results

According to the error matrix in Figure 3, the mT5 model is better at correctly predicting unbiased samples. This is because of the higher unbiased samples in the training data of MAB. In Table 8, the first example for the English CB contains a stereotypical statement "*men are naturally right and it is the role of women to follow their lead*", leading to the correct biased prediction by the model. Similarly, this correct prediction is made in the Swedish CB. We notice over-generalization (May et al., 2019; Nadeem et al., 2021) in the correct examples for the CoLA predictions, where "*every*" is used. The table also shows some incorrect predictions.

5.2 Consistent prediction with perturbation

An interesting property of relative consistency that we observed with the model predictions, as demonstrated with the CoLA dataset, is that when sentences are perturbed, the model mostly maintains its predictions, as long as the grounds for prediction (in this case - over-generalization) remain the same. The perturbations are inherent in the CoLA dataset itself, as the dataset is designed that way. Some examples are provided in Table 9 in the Appendix, where 6 out of 8 are correctly predicted. This property is repeated consistently in other examples not shown here.

5.3 Explainability by graphs

We show explainability by visualization using graphs. Bipol produces a dictionary of lists for every evaluation and we show the *top-5 frequent terms* bar graph for the GNAD10 dataset in Figure 4, which has overall male bias. Many of the 10 evaluated datasets display overall male bias.

5.4 Assumption confirmation through annotation

The results of the annotation of the 200 MAB samples reveal that toxic comments do contain bias. This is shown in Figure 5. The Jaccard similarity coefficient and CUS of IAA are 0.261[6] and 0.515, respectively, given that over 50% is the intersection of unanimous decision.

6 Conclusion

The findings of this work show that bias besets Natural Language Processing (NLP) datasets regardless of language, including benchmark datasets on the GLUE/SuperGLUE leaderboards. We introduced MAB datasets in 3 languages for training models in bias detection. Each has about 2 million labeled samples. We also contribute lexica of bias terms for the languages. In addition, we verified the assumption that toxic comments contain bias. It may be impossible to completely remove bias from data or models, since they reflect the real world, but resources for estimating bias can provide insight into mitigation strategies for reducing bias. Future work may explore ways of minimizing false positives in classifiers to make them more effective. One may also explore how this work scales to other languages or how multilignual models compare to language-specific monolingual models or large language models (LLMs). Regarding culture-specific biases in datasets, one solution will be to

---

[6] Not to be interpreted using Kappa for 2 annotators on 2 classes. Ours involved 3 annotators

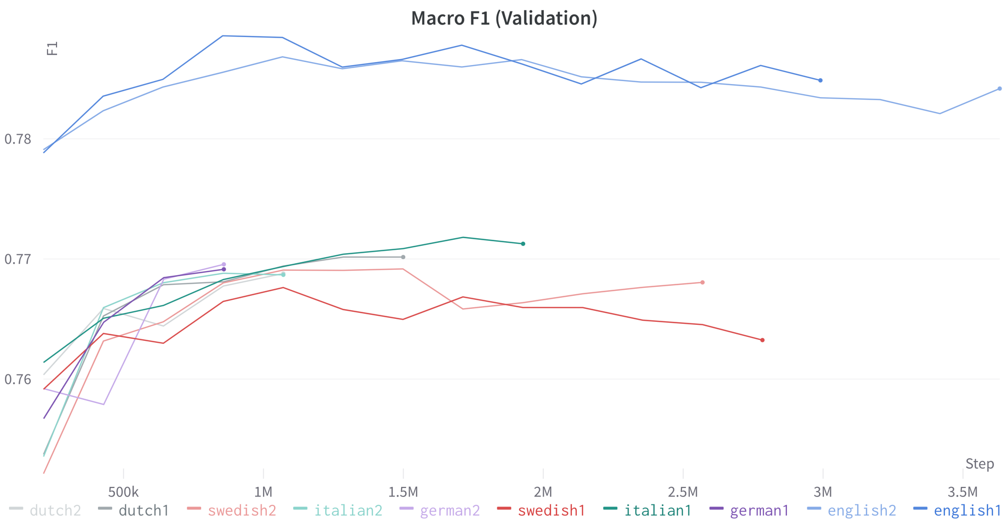


Figure 1: Macro F1 of the validation set for the 5 languages, as generated by wandb.

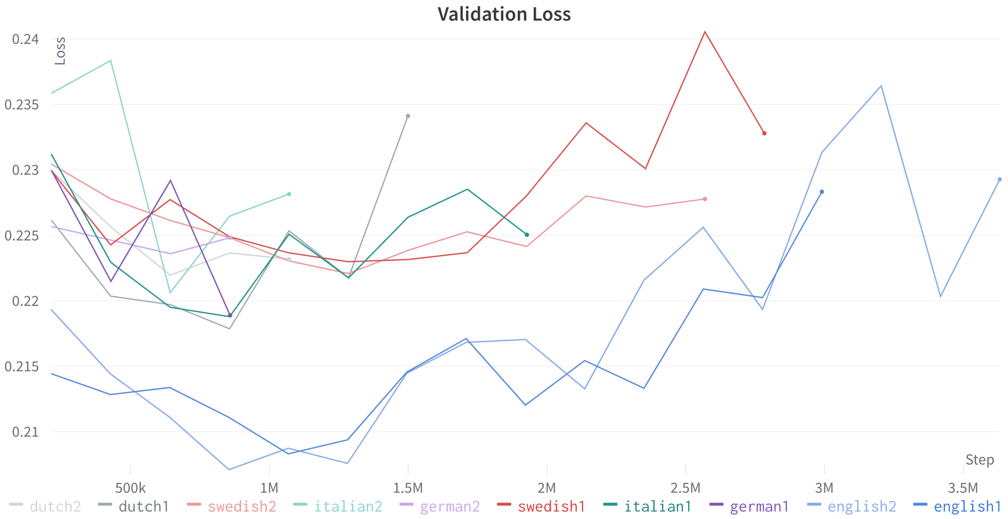


Figure 2: Loss on the validation set for the 5 languages, as generated by wandb.

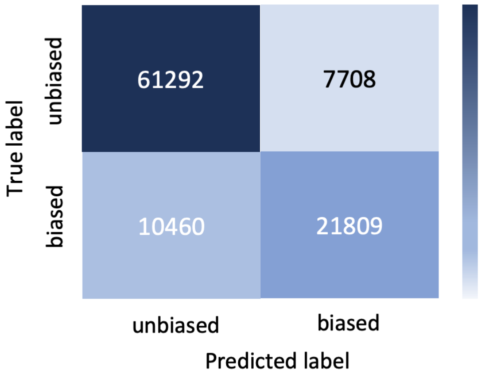


Figure 3: Error matrix of mT5 on MAB-English

collect data from the specific cultures/nationalities to capture these biases.

Ethics statement & limitation

The authors took care while providing examples of data samples, despite some containing stereotypes or toxic content. The classifiers trained for estimating the biases in this work are limited in effectiveness, as shown in the results, hence a result of 0 on any dataset does not necessarily indicate a bias-free dataset. The original MAB was annotated by humans who may have their personal biases based on cultural or demographic backgrounds. This means the final annotations may not be seen as absolute ground truth of social biases.

comment_text	prediction	csv id
Correct predictions from the English CB		
nicky approached her with the assumption that men are naturally right and it is the role of women to follow their lead. constance whose confidence was growing daily was not prepared to give in to nickys wishes merely because of his sex. if she felt he was right then she agreed with him.	biased	85
b: thats true. a: so. b: uh the other argument is that the death penalty is a deterrent and i really dont uh agree with that. i dont think anyone who would commit uh a crime that would get them the death penalty would stop at the moment and say well i was about to kill and dismember this person but oh if they catch me theyre going to kill me so i better not do it. i just dont think uh that it works that way. a: yeah. i dont think its done.	biased	133
Incorrect prediction		
b: yeah and the ground will filter some of it but not all of it. a: no not when you figure i didnt realize one cow produces that much manure	biased	137
Correct predictions from the English CoLA		
if you give him enough opportunity every senator will succumb to corruption.	biased	266
every senator becomes more corrupt the more lobbyists he talks to.	biased	277
Incorrect prediction		
bill squeezed the puppet through the hole.	biased	82

Table 8: Qualitative examples of apparently correct and incorrect predictions in some of the English datasets, based on the mT5 model.

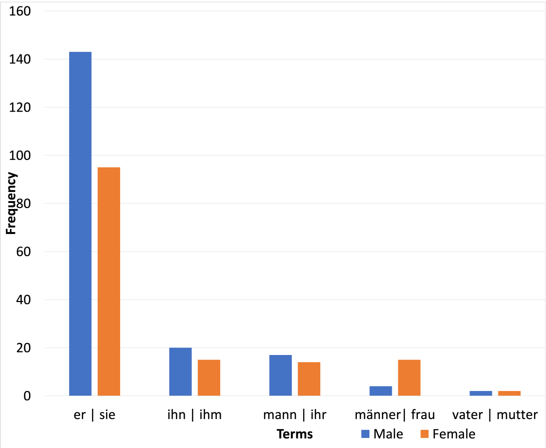


Figure 4: Top 5 frequent terms in the GNAD10 dataset (paired terms are only for comparison).

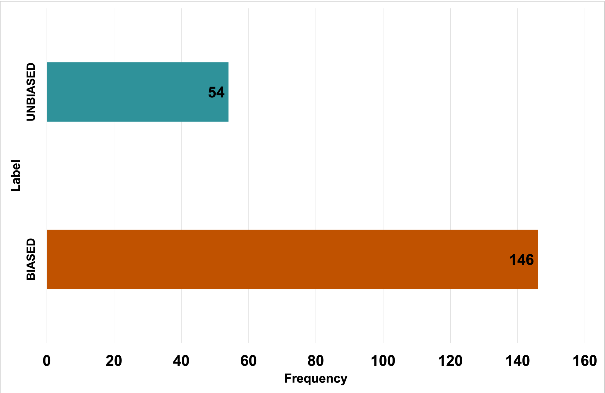


Figure 5: Annotation confirms assumption about toxic comments.

Acknowledgments

The authors wish to thank the anonymous reviewers for their valuable feedback. This work is partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation and counterpart funding from Luleå University of Technology (LTU). The authors also thank Björn Backe for his insightful comments during the writing of this paper.

References

Tosin Adewumi, Mofetoluwa Adeyemi, Aremu Anuoluwapo, Bukola Peters, Happy Buzaaba, Oyerinde Samuel, Amina Mardiyyah Rufai, Benjamin Ajibade, Tajudeen Gwadabe, Mory Moussou Koulibaly Traore, Tunde Oluwaseyi Ajayi, Shamsuddeen Muhammad, Ahmed Baruwa, Paul Owoicho, Tolulope Ogunremi, Phylis Ngigi, Orevaoghene Ahia, Ruqayya Nasir, Foteini Liwicki, and Marcus Liwicki. 2023a. Afriwoz: Corpus for exploiting cross-lingual transfer for dialogue generation in low-resource, african languages. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Tosin Adewumi, Isabella Södergren, Lama Alkhaled, Sana Sabah Sabry, Foteini Liwicki, and Marcus Liwicki. 2023b. Bipol: Multi-axes evaluation of bias with explainability in benchmark datasets. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Varna, Bulgaria.

Lama Alkhaled, Tosin Adewumi, and Sana Sabah Sabry. 2023. Bipol: A novel multi-axes bias evaluation metric with explainability for nlp. *Natural Language Processing Journal*, 4:100030.

Maria Antoniak and David Mimno. 2021. Bad seeds: Evaluating lexical methods for bias measurement. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1889–1904.

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based attention regularization frees unintended bias mitigation from lists. *arXiv preprint arXiv:2203.09192*.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Michael Brownstein. 2019. Implicit Bias. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Fall 2019 edition. Metaphysics Research Lab, Stanford University.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Rodrigo Alejandro Chávez Mulsa and Gerasimos Spanakis. 2020. Evaluating bias in Dutch word embeddings. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 56–71, Barcelona, Spain (Online). Association for Computational Linguistics.

Won Ik Cho, Jiwon Kim, Jaeyeong Yang, and Nam Soo Kim. 2021. Towards cross-lingual generalization of translation gender bias. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 449–457.

Danilo Croce, Alexandra Zelenanska, and Roberto Basili. 2018. Neural learning for question answering in italian. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 389–402, Cham. Springer International Publishing.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Hannah Devinney, 1974 Björklund, Jenny, and Henrik Björklund. 2020. Semi-supervised topic modeling for gender bias discovery in english and swedish. *EQUITBL Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 79 – 92.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Samer El Abassi and Sergiu Nisioi. 2020. Mdd@ ami: Vanilla classifiers for misogyny identification. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 55.

Bo Feng, Qiang Fu, Mianxiong Dong, Dong Guo, and Qiang Li. 2018. Multistage and elastic spam detection in mobile social networks through deep learning. *IEEE Network*, 32(4):15–21.

Antske Fokkens, Nel Ruigrok, Camiel Beukeboom, Gagestein Sarah, and Wouter van Atteveldt. 2018. Studying muslim stereotyping through microportrait extraction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marco Gaido, Beatrice Savoldi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. How to split: the effect of word segmentation on gender bias in speech translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3576–3589, Online. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

Frances M Haemmerlie and Robert L Montgomery. 1991. Goldberg revisited: Pro-female evaluation bias and changed attitudes toward women by engineering students. *Journal of Social Behavior and Personality*, 6(2):179.

Simon Heron. 2009. Technologies for spam detection. *Network Security*, 2009(1):11–15.

Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J Paul. 2020. Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. *arXiv preprint arXiv:2002.10361*.

Styliani Katsarou, Borja Rodríguez-Gálvez, and Jesse Shanahan. 2022. Measuring gender bias in contextualized embeddings. *Computer Sciences and Mathematics Forum*, 3(1).

Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. 2012. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801.

Corina Koolen and Andreas van Cranenburgh. 2017. These are not the stereotypes you are looking for: Bias and fairness in authorial gender attribution. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 12–22, Valencia, Spain. Association for Computational Linguistics.

Anne C Kroon, Damian Trilling, Toni GLA van der Meer, and Jeroen GF Jonkman. 2020. Clouded reality: News representations of culturally close and distant ethnic outgroups. *Communications*, 45(s1):744–764.

Anne C Kroon and Toni GLA van der Meer. 2021. Who's to fear? implicit sexual threat pre and post the "refugee crisis". *Journalism Practice*, pages 1–17.

Mascha Kurpicz-Briki. 2020. Cultural differences in bias? origin and gender bias in pre-trained german and french word embeddings.

Mascha Kurpicz-Briki and Tomaso Leoni. 2021. A world full of stereotypes? further investigation on origin and gender bias in multi-lingual word embeddings. *Frontiers in big Data*, 4:20.

Alessandra Luccioli, Silvia Bernardini, and Raffaella Baccolini. Stereotipi di genere e traduzione automatica dall'inglese all'italiano: uno studio di caso sul femminile nelle professioni.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Katherine McCurdy and Oguz Serbetci. 2020. Grammatical gender associations outweigh topical gender bias in crosslinguistic word embeddings. *arXiv preprint arXiv:2005.08864*.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. 2018. Source-driven representations for hate speech detection. *Computational Linguistics CLiC-it 2018*, page 258.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Rasmus Precenth. 2019. Word embeddings and gender stereotypes in swedish and english.

Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. 2020. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 145–151, New York, NY, USA. Association for Computing Machinery.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Argentina Anna Rescigno, Eva Vanmassenhove, Johanna Monti, and Andy Way. 2020. A case study of natural gender phenomena in translation. a comparison of google translate, bing microsoft translator and deepl for english to italian, french and spanish. In *CLiC-it*.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Magnus Sahlgren and Fredrik Olsson. 2019. Gender bias in pretrained swedish embeddings. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics, NoDaLiDa 2019, Turku, Finland, September 30 - October 2, 2019*, pages 35–43. Linköping University Electronic Press.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan.

Kinshuk Sengupta, Rana Maher, Declan Groves, and Chantal Olieman. 2021. Genbit: measure and mitigate gender bias in language datasets. *Microsoft Journal of Applied Research*, 16:63–71.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Eva Vanmassenhove and Johanna Monti. 2021. gender-it: An annotated english-italian parallel challenge set for cross-linguistic natural gender phenomena. *arXiv preprint arXiv:2108.02854*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Melvin Wevers. 2019. Using word embeddings to examine gender bias in dutch newspapers, 1950-1990. *arXiv preprint arXiv:1907.08922*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

Swedish lexica and the English equivalent

1. Gender-female
   i) hon (she) ii) hennes (her) iii) flicka (girl) iv) mor (mother) v) kvinna (woman) vi) dotter (daughter) vii) mormor (grandmother) viii) dam (lady) ix) sondotter (son's daughter) x) dotterdotter (daughter's daughter) xi) tjej (girl) xii) tjejer (girls) xiii) gumma (old woman) xiv) fru (wife) xv) slampa (slut) xvi) slyna (slut) xvii) lebb (lesbian)

2. Gender-male
   i) han (he) ii) pojke (boy) iii) kille (boy) iv) far (father) v) farfar (grandfather) vi) hans (his) vii) man (man) viii) son (son) ix) make (husband) x) herre (lord) xi) herrar (lords) xii) sonson (son's son) xiii) dotterson (daughter's son) xiv) gubbe (old man) xv) farbro (uncle) xvi) broson (nephew) xvii) män (men)

3. Racial-black
   i) neger (nigger) ii) blåneger (nigger) iii) blåland (nigger) iv) blatte (dark immigrants) v) svartskalle (black head) vi) turk (non-Swedish) vii) sosse (political slur) viii) svarting (blackness) ix) partysvenska (political slur) x) nigga (nigger)

4. Racial-white
   i) svenne (Swedish stereotype) ii) turk (non-Swedish) iii) jugge (derogatory term) iv) sosse (political slur) v) hurrare (Finnish Swedish) vi) lapp (Sami people) vii) Ang mo (red hair) viii) partysvenska (political slur) ix) Ann (White woman) x) rutabaga (Swede)

Example of predictions for the CoLA dataset

comment_text	prediction	csv id
if you give him enough opportunity every senator will succumb to corruption.	biased	266
you give him enough opportunity and every senator will succumb to corruption.	biased	267
we gave him enough opportunity and sure enough every senator succumbed to corruption.	unbiased	268
if you give any senator enough opportunity he will succumb to corruption.	biased	269
you give any senator enough opportunity and he will succumb to corruption.	biased	270
you give every senator enough opportunity and he will succumb to corruption.	biased	271
we gave any senator enough opportunity and sure enough he succumbed to corruption.	biased	272
we gave every senator enough opportunity and sure enough he succumbed to corruption.	unbiased	273

Table 9: Mostly consistent correct prediction with perturbation in the CoLA dataset.

Improved Spoken Emotion Recognition With Combined Segment-Based Processing And Triplet Loss

Dejan Porjazovski and **Tamás Grósz** and **Mikko Kurimo**
Department of Information and Communications Engineering
Aalto University, Espoo, Finland
`dejan.porjazovski@aalto.fi`

Abstract

Traditional spoken emotion recognition solutions often process entire utterances all at once, ignoring the emotional variability within the speech. This shortcoming, especially plaguing end-to-end models, prompted us to investigate a segment-based technique processing only short parts of the audio, improving the recognition accuracy across three diverse emotion datasets. Furthermore, we employed a triplet loss to increase inter-class separability, demonstrating that combining it effectively with segment-based processing within our multi-task learning framework leads to improvements on both English and Finnish datasets. Our proposed method achieves 8.1% unweighted average recall improvement over the baseline on the IEMOCAP, 12% on the RAVDESS, and 7.2% on the FESC dataset. The results also indicate that vocalised emotions are strongly concentrated in short segments of speech, and new methods are needed to exploit this fact.

1 Introduction

In the age of digital transformation, the significance of human-computer interaction (HCI) systems becomes crucial. However, current HCI solutions struggle to comprehend emotions, a critical aspect of tasks like automated analysis of customer feedback. Incorrectly categorising emotions in such analyses could lead to misunderstandings, where complaints might be mistaken for positive feedback and vice versa. Therefore, the integration of an accurate spoken emotion recognition (SER) system within HCI applications holds vital importance in enhancing user experiences (Brave and Nass, 2007).

With the emergence of the Transformer architecture (Vaswani et al., 2017), pre-trained self-supervised models have gained popularity, particularly for tasks with limited data (Grósz et al., 2022). One popular audio-based foundation model is wav2vec2 (Baevski et al., 2020), which has already proven successful in SER applications. In a previous study, the authors utilised a pre-trained wav2vec2 model to extract embeddings from multiple layers, subsequently employing these embeddings as input for a neural network classifier (Pepino et al., 2021). Besides serving as feature extractors, these pre-trained models can also be fine-tuned for the specific task at hand. A fine-tuned wav2vec2 approach was successfully applied for predicting emotional intensities (Porjazovski et al., 2023). In addition to fine-tuning, the researchers incorporated a pre-training stage for the wav2vec2 model, outperforming the other approaches (Chen and Rudnicky, 2023) on the IEMOCAP dataset (Busso et al., 2008).

Despite their popularity, the majority of SER solutions process the whole utterance at once to produce emotion labels. Processing long sequences can cause the model to learn unwanted correlations (Arjovsky et al., 2019). A common way to deal with lengthy audios is to process them in segments (Schuller and Rigoll, 2006; Chen and Rudnicky, 2023; Xia et al., 2021; Tzinis and Potamianos, 2017). We hypothesise that segment-based processing ensures that the model is aware of the varying emotional intensities within the sample, thus improving its accuracy. Moreover, by seeing short segments during training, the model can become more robust to variance in duration. As exact labels for each segment are unavailable, we assigned the same utterance label to all corresponding segments. While not perfect and acknowledging potential label variation across segments, this approach has still proven advantageous (Mao et al., 2020).

The second issue of SER is the limited nature of available data, often addressed by employing unsupervised learning. In a previous study, Trigeorgis et al. (2016) used contrastive predictive coding to learn audio representation in an unsupervised way (Li et al., 2021). Similarly, contrastive loss was used to train a Siamese network (Bromley et al., 1993), which learned to extract discriminative audio features (Lian et al., 2018). Pre-trained transformer models such as wav2vec2 can also benefit from task-specific contrastive learning. In another study, the authors showed the benefits of the wav2vec2 model in combination with contrastive learning and data augmentation (Alaparthi et al., 2022). Closely related to contrastive learning is the triplet loss function (Schroff et al., 2015), which was shown to be beneficial in increasing the inter-class separability of emotions (Huang et al., 2018).

In contrast to prior methodologies that employ the contrastive or triplet loss function across entire utterances, our study introduces a multi-task framework. Here, the model concurrently learns to separate segments with different emotions while optimising the parameters for the SER task using negative log-likelihood loss. By simultaneously applying the loss function at both utterance and segment levels, our approach enables the model to concentrate on both local and global features within the utterances, helping the model understand how emotions change over time. To evaluate the effectiveness of our proposed methods, we conduct empirical experiments using the wav2vec2 model on three distinct datasets in both English and Finnish languages.

2 Datasets

The IEMOCAP dataset contains 12 hours of English speech, annotated with nine discrete emotions. To ensure consistency with prior research, we focused on neutral, sadness, happiness, and anger, omitting the unbalanced classes. To evaluate our solutions, we employed a five-fold cross-validation based on the five recording sessions, as used in other studies (Chen and Rudnicky, 2023; wen Yang et al., 2021).

The second English corpus, called RAVDESS (Livingstone and Russo, 2018), features 12 male and 12 female speakers expressing eight emotions: neutral, calm, happy, sad, angry, fearful, disgust, and surprise, through spoken and sung sentences. As official dataset splits are unavailable, we adopted the splits from Pepino et al. (2021). We merged the calm and neutral emotions and allocated speakers 1-20 for training, 21-22 for development, and 23-24 for testing.

The FESC dataset (Airas and Alku, 2006) comprises Finnish prose passages narrated by five male and four female actors, spanning five hours. The dataset contains annotations for neutral, sadness, joy, affection, and anger emotions. In our experiments, we prepared the data the same way as done by Vaaras et al. (2022) employing a leave-one-speaker-out cross-validation approach, with each fold featuring one speaker for testing, one for validation, and the rest for training. The monotonic character of Finnish, primarily resulting from minimal pitch variation and placement of stress on the first syllable of the words, poses unique difficulties for emotion recognition.

3 Methods

Our proposed model utilises a multi-task setting, optimising two objectives. The negative log-likelihood helps the model learn to successfully classify the emotions, while the triplet loss separates the utterances with different labels farther in the latent space. Moreover, to reduce the variance attributed to varying lengths, we additionally use segment-based processing within the triplet and negative log-likelihood losses. The proposed model is illustrated in Figure 1.

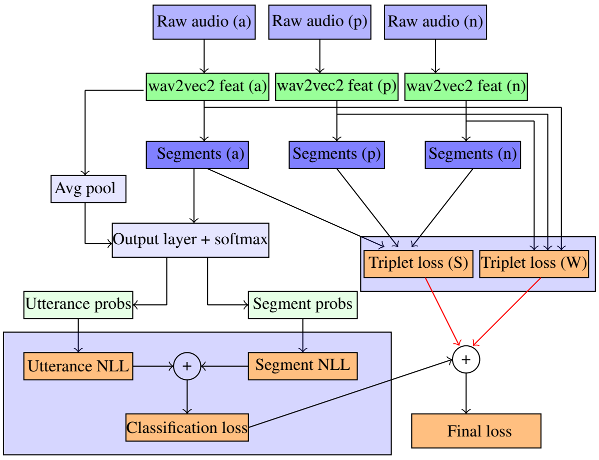


Figure 1: Architecture of the proposed model. "a" refers to the anchor, "p" to the positive, and "n" to the negative element. During training, we either use the triplet loss on the segments (S) or on the whole utterance (W)

3.1 Segment-based processing

To extract features from the raw audio sequence, we used a wav2vec2 model. The embedded data are of shape $X = (N, T, H)$, where N is the batch size, T is the temporal dimension, i.e. the timesteps, and H is the hidden size. As discussed earlier, the goal is to process the utterance in small segments.

We split each embedding vector X into segments with overlapping windows. Then, we average each segment along its temporal dimensions and pass it to a Linear layer, followed by a Softmax function, which produces class probabilities. This way, the model will generate label probabilities for each segment. In case the temporal dimension of the embedding vector X is smaller than the segment size, we process the whole sequence at once without splitting it. During training, we compute the loss over the whole sequence, as well as over each of the segments. In the inference stage, we obtain the label prediction by selecting the segment containing the highest probability.

3.2 Triplet loss

Our multi-task loss function is defined as:

$$L = L_{nll} + L_{tri} \tag{1}$$

where L_{nll} is the negative log-likelihood loss. The triplet loss function L_{tri} is calculated as:

$$L_{tri} = max(d(X_a, X_p) - d(X_a, X_n) + \lambda, 0) \tag{2}$$

where X_a is the anchor element, X_p is the positive element from the same class as X_a, and X_n is the negative element from a different class. The goal of the triplet loss function is to make the distance d between the elements of the same classes smaller than the distance between the elements of different classes. The distance d, in our study, is the L2 norm. λ is a margin determining the minimum distance between the positive (X_a, X_p) and the negative pairs (X_a, X_n). For choosing the negative sample, we ordered the samples by length and chose them to have a similar duration as the positive ones. This was done so that we would not compute the distance between a whole utterance that can not be split and a segment. There are other viable methods for selecting the negative sample, for instance, by picking one with a different valence or arousal; however, that is application-specific and we do not consider it in this study.

As discussed earlier, processing the utterance in segments can reduce the variance attributed to varying lengths. The triplet loss, on the other hand, helps with pulling the latent representation of samples with different classes farther from each other, ensuring easier separability. Therefore, we combined segment-based processing and triplet loss to utilise the benefits of both.

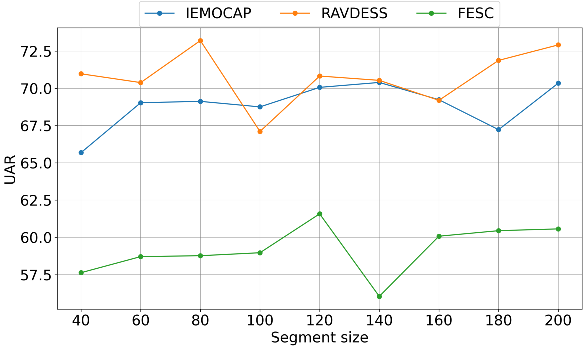


Figure 2: The effect of the segment size. The stride is half of the segment size. A segment size of 100 refers to roughly 200ms.

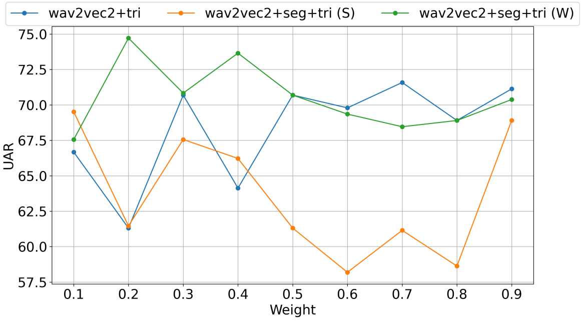


Figure 3: The effect of the weight α when combining the negative log-likelihood and triplet loss for the RAVDESS dataset.

To learn the SER task, we used the negative log-likelihood loss function on each segment, as well as the whole utterance:

$$L_{SER} = \sum_{i=1}^{N} \sum_{j=1}^{S} L_{nll}(i,j) + L_{nll}(i) \tag{3}$$

where N is the number of samples, i is the sample, S is the number of segments in sample i, j is the segment, and (i, j) represents the j-th segment in sample i.

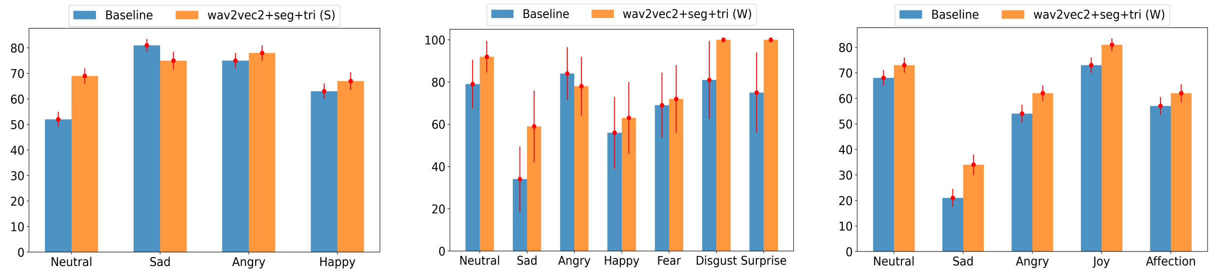
In the experiments, we used either the segmented or the whole utterance in the triplet loss. In the end, we interpolated both loss functions.

4 Experiments

To extract features from the FESC utterances, we employed the multilingual pre-trained wav2vec2 model (Conneau et al., 2021), fine-tuned for ASR on Finnish data[1] (~311M trainable parameters). In this study, we did not consider other self-supervised models, like HuBERT (Hsu et al., 2021) or WavLM (Chen et al., 2022), since they do not

---

[1] jonatasgrosman/wav2vec2-large-xlsr-53-finnish



(a) IEMOCAP (b) RAVDESS (c) FESC

Figure 4: UAR per class and the 95% confidence intervals (in red) for the baseline and the best model on each dataset.

have a Finnish variant. For the English experiments, we utilised the base wav2vec2 version[2] (∼90.2M trainable parameters), which is not fine-tuned on any specific task. The feature dimensions were set at 1024 for Finnish and 768 for English. We extracted English wav2vec2 features from the last Transformer layer, while for the Finnish version, we utilised layer 23 (out of 24), given that the final layer is typically optimised for the ASR task (Pepino et al., 2021). Even though performing a layer analysis and choosing the best-performing one can potentially improve the results, in this study, we focus on the architecture instead of specific hyperparameters.

To select the optimal segment and stride sizes, we tested the performance of the models with different values. The results of this experiment are shown in Figure 2. For the datasets where we employ cross-validation, we determined the best segment size on one fold. Based on the figure, on the IEMOCAP dataset, a segment size of 140 with a stride of 70 was chosen as the most optimal. For RAVDESS, smaller segment and stride sizes of 80 and 40 gave the best results, whereas, for FESC, a segment size of 120 with a stride of 60 performed the best.

The margin value λ in Equation 1 was set to 1 in all the experiments. When combining the negative log-likelihood and triplet loss functions, we did not use a weighting factor for IEMOCAP and FESC datasets. This decision was based on the high-performance variability between folds; the most optimal value for some splits resulted in poor outcomes for others. To address this, we attempted to set the weight as a learnable parameter, but this did not yield better results compared to not using any weighting.

For the RAVDESS dataset, since we did not use cross-validation, we conducted a weight analysis to determine the optimal value, as shown in Figure 3. We performed the weight analysis on a subset of the training set and determined the best weight based on the development set. To factor the weight when combining the loss functions as in Equation 1, we used:

$$L = (1 - \alpha) * L_{nll} + \alpha * L_{tri} \tag{4}$$

The weighting analysis revealed that the most optimal α value was 0.7 for the wav2vec2 model utilising the triplet loss, 0.1 for the model combining segmented processing and triplet loss on the segments (S), and 0.2 for the model using the triplet loss on the whole utterance (W). These findings revealed that when using segmented processing, it is better to give more weight to the negative log-likelihood loss, while when processing the whole utterance, it is better to give more priority to the triplet loss.

For optimisation, we used the Adam optimiser and trained the models for 30 epochs using a single V100 GPU. For the most complex model that uses segmented processing and triplet loss during training, the training time for one epoch with a batch size of 12 took around 34 minutes. During training, we kept the Convolutional Feature Encoder frozen while fine-tuning the Transformer layers. The complete code, along with a detailed list of hyperparameters, is publicly available[3].

5 Results

In this section, we compare the proposed techniques against the standard wav2vec2 pipeline, processing the whole utterance at once. We used unweighted average recall (UAR) as an evaluation

---

[2] facebook/wav2vec2-base

[3] Removed due to anonymity

Model	IEMOCAP	RAVDESS	FESC
wav2vec2 P-TAPT (Chen and Rudnicky, 2023)	(74.3)	/	/
wav2vec2+layer avg (Pepino et al., 2021)	67.2	84.3	/
wav2vec2 baseline	66.5 (65.6)	68.5 (67.8)	57.1 (60.5)
wav2vec2+seg	67.6 (66.7)	73.2 (72.1)	60.1 (62.1)
wav2vec2+tri(W)	73.6 (72.6)	80.4 (78.4)	61.0 (62.1)
wav2vec2+seg+tri(S)	**74.6 (73.9)**	79.0 (76.9))	63.6 (64.9)
wav2vec2+seg+tri(W)	73.9 (72.9)	**80.5 (78.4)**	**64.3 (65.0)**

Table 1: UAR and UA (given in the brackets) scores for the IEMOCAP, RAVDESS, and FESC test sets. (S) indicates that the triplet loss was calculated on the segments, while(W) indicates that it was done on the whole utterance.

metric. The UAR metric is calculated as a sum of the class-wise recall divided by the number of classes. For comparison with the previous state-of-the-art (SOTA) method on IEMOCAP, we additionally provide unweighted accuracy (UA) scores.

Table 1 presents the UAR and UA scores achieved on the IEMOCAP, RAVDESS, and FESC datasets. Looking at the IEMOCAP test results, we can notice that fine-tuning the wav2vec2 model with a classification layer already yields good performance. Segment-based processing, which involves splitting the utterances into segments and processing them individually, slightly improves the results over the wav2vec2 baseline.

To explore the impact of segment-based processing on recognising sequences of varying lengths, we divided the test set into segments below and over 10 seconds in duration. For short utterances, the baseline achieved a recognition rate of 65.9%, whereas segment-based processing notably improved the performance to 67.2% UAR, underscoring its efficacy for shorter utterances. Conversely, for utterances longer than 10 seconds, segment-based processing got slightly inferior results with 69.7% UAR, compared to the baseline's 70.6%. These findings highlight the nuanced effect of segment-based processing, demonstrating its effectiveness for short sequences while indicating the need for further optimisation or alternative approaches for longer ones.

Introducing the triplet loss, combined with negative-log-likelihood, gives a further improvement of 7.1% UAR score over the wav2vec2 baseline. Furthermore, we observed additional improvement by combining segment-based processing and triplet loss. When using the segments in the triplet loss, the model got a 74.6% UAR score.

On the RAVDESS test set, the segment-based processing achieved a UAR score of 73.2%, considerably better than the baseline of 68.5%. Using the triplet loss further enhances the results over solely using the segmented processing. The multi-task learning approach produces the best UAR score of 80.5%, this time by using the whole utterance in the triplet loss. Since the lengths of the utterances in this dataset are short, we could not assess the performance of the segment-based model on short and long samples.

The Finnish experiments follow a similar trend, where the segment-based processing outperforms the baseline. To examine the impact of segment-based processing on utterance length, we partitioned the test set into segments shorter and longer than 10 seconds, mirroring our approach in the IEMOCAP dataset. Notably, this analysis revealed enhancements in recognition performance for both short and long utterances through segmented processing. Specifically, for short utterances, the baseline wav2vec2 model achieved a UAR score of 57%, while segment-based processing improved it to 59.3%. For longer segments, the difference is more pronounced, with the baseline yielding a UAR score of 69%, contrasted with 72.7% for segment-based processing.

Adding the triplet loss further improves the results, achieving a UAR score of 61%. In the multi-task scenario, employing the triplet loss on entire utterances rather than segments gives better results, as seen from Table 1.

The superior performance of our proposed multi-task model comes at a cost of increased computational time. For instance, to evaluate one split of the IEMOCAP test set, the baseline model took 30 seconds using a batch size of 1, whereas the wav2vec2+seg+tri(S) took 41 seconds.

Compared to SOTA results on the IEMOCAP dataset, our multi-task model using triplet loss, in combination with segmented processing, achieves

Model	IEMOCAP	RAVDESS	FESC
wav2vec2 baseline	75.1	78.3	71.5
wav2vec2+seg	74.5	78.2	72.8
wav2vec2+tri(W)	82.8	86.8	80.2
wav2vec2+seg+tri(S)	/	89.7	79.7
wav2vec2+seg+tri(W)	83.0	/	/

Table 2: Model agreement in terms of UAR, where the best model's predictions for each dataset are treated as ground truth.

a slightly worse UA score than the P-TAPT, which modifies the pre-training stage of the wav2vec2 model to generate emotion-specific features. On the RAVDESS dataset, the SOTA results incorporate a weighted average of all wav2vec2 layers, whereas we only utilise the output of the last Transformer layer. Exploring multiple layers or selecting the best layer for the task could potentially improve the results, but this falls beyond the scope of our study. Additionally, that approach performs the best on the RAVDESS dataset, but its performance drops on IEMOCAP, indicating that it is not robust enough. For the Finnish FESC dataset, we could not find a suitable benchmark.

To get a better understanding of the improvements gained from the multi-task model, we plotted the UAR per class for the baseline and the best-performing model on each dataset, shown in Figure 4. Upon examining the class-specific performances of both models across various emotions, it becomes evident that the multi-task approach almost always achieves superior recognition rates. Notably, exceptions include the recognition of sadness in the IEMOCAP dataset and anger in the RAVDESS dataset. A plausible explanation for the diminished performance in recognising the sad emotion might stem from its extended average duration. As previously discussed, the model demonstrates a slight decline in performance when processing long utterances within the IEMOCAP dataset.

To test the stability of the models, we calculated the 95% confidence intervals for the best models on each dataset, using the bootstrapping method. The confidence intervals are presented in Figure 4. For the RAVDESS dataset, we observed a large interval which contains the real performance with a 95% chance. These findings indicate that there is a high variability between the utterances for some of the emotions. Moreover, our model tends to be more stable with less variability for disgust, surprise and neutral emotions in comparison to the baseline. Nevertheless, the performance per class is in the middle of the confidence intervals, meaning that the overall performance is not distorted by some extremely easy or difficult test samples.

In the last set of experiments, we test how much the models differ in the predictions. To achieve that, we calculated the model agreement, where we treated the best-performing model's predictions as ground truth and evaluated it against the other models. The results of this experiment are presented in Table 2.

On the IEMOCAP and RAVDESS datasets, the best-performing model has the biggest agreement with its similar counterpart (wav2vec2+seg+tri(W) for IEMOCAP and wav2vec2+seg+tri(S) for RAVDESS), followed by the model just utilising the triplet loss. On the FESC dataset there is a higher agreement between the best-performing model and the one that only incorporates triplet loss, even though that model falls behind in terms of UAR, compared to both multi-task approaches. These results indicate that the mistakes that both the wav2vec2+seg+tri(S) and wav2vec2+seg+tri(W) models make differ from each other, suggesting that they learn different things when using segments or whole utterances in the triplet loss.

6 Conclusion

In this work, we investigated segment-based processing, triplet loss, and a multi-task combination of both techniques for SER. The results from our English and Finnish experiments demonstrated the effectiveness of segment-based processing compared to the conventional approach of processing the entire utterance at once. Moreover, we showed that the segment size plays an important role and should be chosen carefully. By integrating the triplet loss into the learning framework, we observed considerable performance improvements across all datasets, surpassing the segment-based processing and showing the benefits of separating the different classes in the latent space. On all three corpora, the multi-task approach of combining the segmented processing and triplet loss gave the best results. Furthermore, we showed that segment-based processing improves the model's robustness on short utterances, whereas for long ones, there is a performance drop on the IEMOCAP, but an improvement on FESC. By comparing the model agreement, we found that using segments or whole utterances in the triplet loss can lead to the models learning different things, making their predictions differ.

Limitations

While the proposed multi-task approach significantly outperforms the baseline, it comes with increased computational demands. Further enhancements can be achieved by combining multiple Transformer layers, as demonstrated in (Pepino et al., 2021). However, this study omits layer experimentation to prioritise architectural analysis over hyperparameter tuning. Additionally, not incorporating a weighting factor for IEMOCAP and FESC when combining the loss functions adds a limitation, which remains an important future task.

Ethics Statement

The use of emotion recognition in certain applications can lead to human rights violations. The EU AI act (EU, 2024) has classified emotion recognition systems as a high-risk application, meaning that the users need to be informed if such a system is being put into place. Moreover, relying on automatic emotion recognition systems to determine the state of a person can be dangerous, especially when that person is in shock and may not express the actual emotions. Furthermore, emotional expression varies significantly from person to person. For instance the emotional expression in children with autism differs from typically developing children (Chaidi and Drigas, 2020). While the development of emotion recognition technology may offer societal benefits, it is essential to carefully consider who the primary users are and how they will be affected.

Acknowledgements

The computational resources were provided by Aalto ScienceIT. We are grateful for the Business Finland project LAREINA under Grant 7817/31/2022.

References

Matti Airas and Paavo Alku. 2006. Emotions in vowel segments of continuous speech: analysis of the glottal flow using the normalised amplitude quotient. *Phonetica*, 63(1):26–46.

Varun Sai Alaparthi, Tejeswara Reddy Pasam, Deepak Abhiram Inagandla, Jay Prakash, and Pramod Kumar Singh. 2022. Scser: Supervised contrastive learning for speech emotion recognition using transformers. In *2022 15th international conference on human system interaction (HSI)*, pages 1–7. IEEE.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, et al. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, et al. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Scott Brave and Cliff Nass. 2007. Emotion in human-computer interaction. In *The human-computer interaction handbook*, pages 103–118. CRC Press.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, et al. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Irene Chaidi and Athanasios Drigas. 2020. Autism, expression, and understanding of emotions: literature review.

Li-Wei Chen and Alexander Rudnicky. 2023. Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In *Proc. Interspeech 2021*, pages 2426–2430.

EU. 2024. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts.

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CONSIL%3AST_7536_2024_INIT&qid=1716543737061.

Tamás Grósz, Dejan Porjazovski, Yaroslav Getman, et al. 2022. Wav2vec2-based paralinguistic systems to recognise vocalised emotions and stuttering. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7026–7029.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, et al. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Jian Huang, Ya Li, Jianhua Tao, et al. 2018. Speech emotion recognition from variable-length inputs with triplet loss function. In *Interspeech*, pages 3673–3677.

Mao Li, Bo Yang, Joshua Levy, et al. 2021. Contrastive unsupervised learning for speech emotion recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6329–6333. IEEE.

Zheng Lian, Ya Li, Jianhua Tao, et al. 2018. Speech emotion recognition via contrastive loss under siamese networks. In *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, pages 21–26.

Steven R Livingstone and Frank A Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

Shuiyang Mao, P.C. Ching, C.-C. Jay Kuo, et al. 2020. Advancing Multiple Instance Learning with Attention Modeling for Categorical Speech Emotion Recognition. In *Proc. Interspeech 2020*, pages 2357–2361.

Leonardo Pepino, Pablo Riera, and Luciana Ferrer. 2021. Emotion Recognition from Speech Using wav2vec 2.0 Embeddings. In *Proc. Interspeech 2021*, pages 3400–3404.

Dejan Porjazovski, Yaroslav Getman, Tamás Grósz, and Mikko Kurimo. 2023. Advancing audio emotion and intent recognition with large pre-trained models and bayesian inference. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9477–9481.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Björn Schuller and Gerhard Rigoll. 2006. Timing levels in segment-based speech emotion recognition. In *Proc. Interspeech 2006*, pages paper 1695–Wed2BuP.8.

George Trigeorgis, Fabien Ringeval, Raymond Brueckner, et al. 2016. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE.

Efthymios Tzinis and Alexandras Potamianos. 2017. Segment-based speech emotion recognition using recurrent neural networks. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 190–195. IEEE.

Einari Vaaras, Manu Airaksinen, and Okko Räsänen. 2022. Analysis of Self-Supervised Learning and Dimensionality Reduction Methods in Clustering-Based Active Learning for Speech Emotion Recognition. In *Proc. Interspeech 2022*, pages 1143–1147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, et al. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pages 1194–1198.

Yangyang Xia, Li-Wei Chen, Alexander Rudnicky, et al. 2021. Temporal context in speech emotion recognition. In *Interspeech*, volume 2021, pages 3370–3374.

EEG Signal Analysis for Multimodal Simple Concepts Decoding

Sergio Guillén, Lorenzo J. Tardón, Ana M. Barbancho, and **Isabel Barbancho**
ATIC Research Group, ITIS Software
Universidad de Málaga, Málaga, Spain
`{sergiogj,lorenzo,abp,ibp}@ic.uma.es`

Abstract

In this paper, we explore the use of a feature extraction model for the detection of basic decision-making concepts such as “yes” and “no” in several communication channels using electroencephalography (EEG) signals. Power topographic distribution of both concepts are explored, showing similar pattern activation in all communication channels chosen. Bi-LSTM model was used for the classification of the feature matrices extracted from EEG trials when transformed using real cepstrum, achieving, on average, 81% of accuracy across all subjects. This could help people with disabilities improve their quality life by enabling communication even when vocal communication is not possible.

1 Introduction

Electroencephalography (EEG) has prompted researcher’s attention for years since allowing to record brain activity non-invasively with high temporal resolution (Luck, 2014). Language, as we commonly understand it, is the sole province of humans (Patel, 2007). Phonemes and syllables of human language are acoustically complex entities to produce (Patel, 2007), which historically led researchers to focus on the processing of individual words (lexical items) (Petersen et al., 1988). By locating and understanding the role that each brain area plays in language processing could help people with severe neurological impairments, including communication, such as pure dyslexia or aphasia (Petersen et al., 1988). Historically, their location was obtained with functional neuroimaging techniques such as functional Magnetic Resonance Imaging (fMRI), while their timing was captured by using electromagnetic techniques such as electroencephalography (EEG) (Price, 2012). The use of this non-invasive technique, could provide a means of communication with impaired people when using monosyllable words or simple concepts (Lazarou et al., 2018). By identifying brain activation patterns associated with different communication channels and classifying opposing simple concepts, such as "yes" and "no", when presented in all of them, it becomes possible to establish a direct communication pathway allowing their decoding into words by utilizing electrodes placed in specific brain regions, offering individuals the ability to express themselves without the need for physical speech.

Language processing has been widely explored by researchers, and it has shown to involve several brain regions, with the most popular neural model of language being based on the writings of Broca, Wernicke and Lichtheim at the end of the 19th Century and Geschwind in the mid 20th Century (Price, 2012). The Broadman areas named after them are classically related to language production and processing (Hall and Hall, 2021). However, in recent years, it has been proven the role of more neural structures in language processing (Nizara, 2018). In both (Rezazadeh Sereshkeh et al., 2017; Choi and Kim, 2019), the decoding of “yes” and “no” concepts from EEG signals was achieved through a feature extraction stage, followed by classification. In the former, a multilayer perceptron (MLP) was employed, achieving 63.17% accuracy scores on average, while the latter used a support vector machine (SVM) with a 86.03% attained when combining multiple time-frequency subwindows.

Historically, cepstrum has its roots in the general problem of signals deconvolution (Childers et al., 1977), but it has proved it usefulness, not only in speech signal processing, but also in EEG signal processing (Sen et al., 2023; Han et al., 2024). Inspired from the success of cepstral features we propose a feature extraction model using EEG signal analysis to discern between two monosyllable words with opposite meanings, “yes” and “no”, that could enhance communication possibilities for persons with some kind of motor/neurological disability supported by the use of a Bidirectional Long-Short Term Memory (Bi-LSTM) Neural Network model.

This manuscript is organized as follows: In Sec. 2, the methodology followed to record the EEG signal analyzed is presented. Sec. 3 describes the main pre-processing stages considered and Sec. 4 shows the features extracted that build the features matrix used in the classification stage. Results and conclusions are presented in Secs. 5 and 6, respectively.

2 Methods

In this section, the methodology followed to record EEG signals is described, including the participant's details, experiment description, and the specific equipment used.

2.1 Participants

We recorded data from a heterogeneous group of 15 healthy participants (3 male participants, mean age = 24.00; 12 female participants, mean age = 28.16), most of them members of Universidad de Málaga (UMA), whose participation was entirely voluntary with no monetary or any other kind of compensation. All of them received detailed oral information about the experiment before providing their written informed consent to take part in the experiment.

This study was conducted in accordance with the Declaration of Helsinki and was approved by Comité Ético de Experimentación de la Universidad de Málaga, Reg. CEUMA: 61-2021-H. The privacy and confidentiality of the participants were strictly protected throughout the study.

2.2 Experiment Description

At present, it is not clear that comprehension of a word necessarily entails activation of a detailed perceptual representation of the object to which it refers, at least not to the same degree as that evoked by the object itself (Binder et al., 2009). Inspired by that, in order to detect each word (lexical item/concept) processing, subjects were presented with blocks of "yes" and "no" words in the same language (Spanish), presented in different ways, we will refer to them from now on as communication channels. These are aimed to cover all possible ways of communication, including these scenarios:

- **Read:** Words were displayed as text on the screen, so participants were instructed to read them.
- **See:** Words were displayed with a representative symbol on the screen, so participants were instructed to look at them.
- **Listen:** Words were played through the speakers, so participants were instructed to listen to them.
- **Say:** Words were displayed as text on the screen, so participants were now instructed to read them out loud.
- **Think:** Words were shown as text on the screen, so participants were now instructed to think of the word displayed.

In the two last scenarios, each word was first presented and then, the action was performed, expecting to avoid the mixture of different cognitive stimuli. Note that although the appearance of these blocks was not randomized, the presentation of the words within them, lasting 7 seconds on average, was.

Recall that all participants recruited were healthy subjects. Because of this, the experiment is designed so that no specific feedback is necessary from subjects during the experiment's recording. Also, some resting time is allocated between sub-experiments.

Experiments were conducted in a separate room with soundproof windows, and curtains to avoid external noise. Non-essential electronic devices were turned off to reduce electromagnetic interference. Participants were instructed to avoid unnecessary movements.

2.3 Equipment

BrainVision's actiChamp-Plus and acti-CAP were used (Brain Products, 2016) in this work. 64 active electrodes were arranged according to the $10-20$ system (American, 1994; Klem et al., 1999), which provide high-quality recordings with low background noise. Among them, FCz and FPz are used as reference and ground channels, respectively, while FT9 and FT10 electrodes are displaced to record vertical (VEOG), and horizontal (HEOG) ocular activity. Iz electrode is used for low-quality audio capture ($f_s = 2500 Hz$), leading to a final count of 61 electrodes used to measure EEG signals. The maximum impedance measured across all participants was kept under $10k\Omega$, and balanced during every recording session (Sanei and Chambers, 2013). Figure 1 shows the electrode positioning configuration employed.

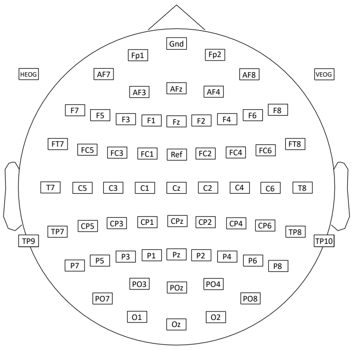


Figure 1: Electrode locations used.

BrainVision Recorder (Brain Products, 2021) was used to capture EEG data supported by E-Prime (W. Schneider and Zuccolotto, 2016) running on an external computer for stimuli presentation, and sending timestamps to the EEG recorder. Speakers connected to the E-Prime computer allowed for audio stimuli presentation. Their volume was kept constant to avoid differences between participants.

3 EEG data Pre-processing

This section details the pre-processing steps taken using MATLAB R2022a (MathWorks, 2022), along with Fieldtrip Toolbox (Oostenveld et al., 2010) for EEG data handling.

EEG signal's small amplitude requires them to be amplified in order to be properly analyzed (Luck, 2014). This causes several noise sources, both correlated and uncorrelated, to be also amplified, potentially masking the neural activity of the brain (Cohen, 2014). To address this issue, both a high-pass filter and low-pass filter, with cut-off frequencies of $0.1Hz$ and $45Hz$, respectively, were applied to eliminate slow drifts and non-cognitive signals, also proving a baseline correction for each subject. Also, considering muscle artifacts falling within the $30 - 100Hz$ frequency range (Luck, 2014), the chosen cut-off frequencies should effectively mitigate them (Hassan and Hussain, 2023).

After filtering, EEG signals are down-sampled to $f'_s = 100Hz$ to reduce the computational cost without compromising the results.

Independent Component Analysis (ICA) is then applied for visual artifact rejection (Sanei and Chambers, 2013). This technique decomposes electrode signals, $\boldsymbol{x}(k) = [x_1(k), x_2(k), \ldots, x_e(k)]^T$, into statistically independent components (IC), through and unmixing matrix, $\boldsymbol{W}$, according to:

$$\boldsymbol{i}(k) = \boldsymbol{W} \cdot \boldsymbol{x}(k) \tag{1}$$

where $\boldsymbol{i}(k)$ are statistically independent signal components. Electrooculogram (EOG) and audio channels were removed for IC extraction. Note that k refers to the samples of the signal within the excerpt considered.

Based on the approach presented in (Villena et al., 2019), a threshold process based on the correlation coefficients was applied to detect potential artifactual components and discard them before mixing the remaining ICs back. Both the ICs obtained and the EEG signal are pre-epoched in the segments of interest (trials), and the Pearson Correlation Coefficients (ρ) were obtained by following:

$$\rho(e,n) = \frac{\sum_{k=1}^{K}(x_e(k)-\overline{x_e})\big(i_n(k)-\overline{i_n}\big)}{\sqrt{\sum_{k=1}^{K}(x_e(k)-\overline{x_e})^2 \sum_{k=1}^{K}\big(i_n(k)-\overline{i_n}\big)^2}} \tag{2}$$

where e refers to each EOG channel considered, and n to the n-th component extracted.

Note that this step is performed for each selected trial, and components are considered as artifactual when they surpass the defined threshold, in its absolute value, in at least the 80% of the trials considered. To avoid erasing cognitive information, this threshold process was supported by visual supervision of the detected components, ensuring that only artifactual information was removed from the original signal (non epoched) $x_e(k)$, to obtain the reconstructed signal, $x'_e(k)$.

Since in this experiment we are interested in assessing neural processing of different words or concepts, EEG signals are now epoched to contain the two words considered, i.e., “yes” and “no” in all the communication channels assessed. With this in mind, the power topographic distribution of all subjects is compared in all the scenarios assessed. Figure 2 shows the scalp power distribution averaged for all participants available when presented with “yes” and “no” words in each communication channel. Note that for each channel of communication, the topographic distribution of both words are displayed using the same normalized color axis.

In this figure, it can be observed how some activation areas are present in all communication channels chosen when processing these two concepts. Primary, the main activity is focused on the frontal

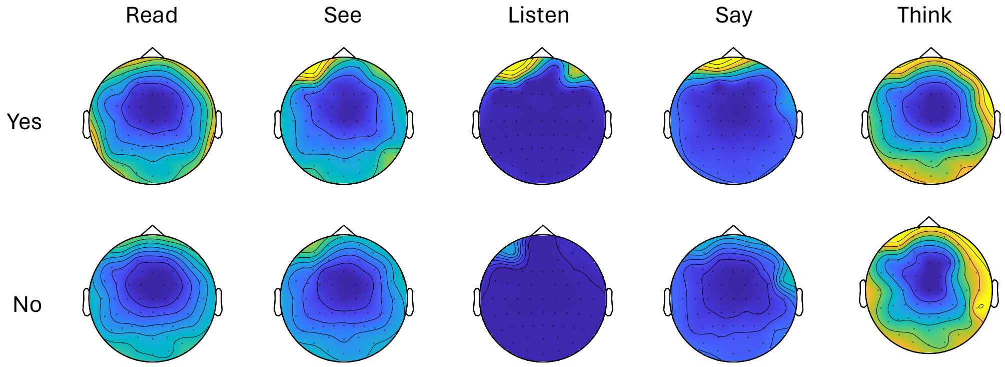


Figure 2: Average scalp power distribution when processing "yes" and "no" words/concepts when different communication channels are used.

region rather than parietal-occipital and temporal, depending on the task performed.

As for the former, mainly a left-laterallyzed activity over the frontal cortex is consistent with what is stated in (Petersen et al., 1988), where task calling for semantic processing of individuals words activation was observed over the frontal region. Later neuro-psychological experiments showed that damages in frontal or inferior parietal areas in this hemisphere caused deficits in tasks that required speech syllables identification, which highlighted the possible role of a fronto-parietal circuit in the perception of speech (Hickok and Poeppel, 2007). However, note that is not always restricted to the left hemisphere. This is not odd since previous researches indicates that there is probably, at least, one pathway in each hemisphere related with speech processing (Hickok and Poeppel, 2007). This behavior is also consistent with the spatial distribution observed in (Choi and Kim, 2019), where right frontal region exhibited the most useful features for the discrimination task.

There is strong evidence that posterior middle temporal regions are involved in accessing lexical and semantic information. Although it's not present with the same magnitude in all cases assessed, activation in left temporal region directly relates with Broca's and Wernicke's areas, both intimately related with language processing (Petersen et al., 1988; Hickok and Poeppel, 2007). Semantic processing has been particular observed over inferotemporal and posterior inferior parietal regions, as stated in (Binder et al., 2009). Activation in these regions are somehow expected since posterior areas are involved in visual feature extraction and more anterior areas are involved in lexico-semantic processing of the whole word (Price, 2012).

Language processing is widespread and occupy a large proportion of the cortex in the human brain (Binder et al., 2009), with its neural organization, being task dependent, as stated in (Hickok and Poeppel, 2007).

At the view of these power topographic distributions, we hypothesized that the processing of both concepts is equivalent regardless of the type of medium chosen for their presentation. Based on this, they will be treated equally without discerning between them in further stages.

For concept processing detection, the real cepstrum is applied to the original EEG signal. Its used is based on previous results as presented in sec. 1. The real cepstrum of a signal is obtained as the Inverse Fourier Transform of the logarithm of the magnitude of the spectrum (Shourie, 2016). In our research, the real cepstrum analysis process has been applied to the EEG signals to use their coefficients as a parameter vector to characterize signals and analyze them improving the results obtained when compared with the use of the EEG signal directly. The real cepstrum can be computed as:

$$c_{e,t}(k) = \frac{1}{2\pi} \int log|X_{e,t}(w)|e^{jwk}dw \qquad (3)$$

where $c_{e,t}(k)$ are the real-valued coefficients of the cepstrum for the e-th electrode, and the t-trial, and $X_{e,t}(w)$ the Fourier transform of the input signal, $x'_{e,t}(k)$.

4 Feature Extraction

This step allows for reducing the complexity of the classification step (Alghamdi et al., 2023; Danyal et al., 2023). To this end, each feature was extracted following a single-trial method, i.e., within each subject, trial, and electrode (Grierson and Kiefer, 2014).

Based on previous EEG research (Hassan and Hussain, 2023; Al-Qazzaz et al., 2023), once individually tested, both statistical and Power Spectral Density (PSD) features-based are chosen to maximize the results obtained when combined:

- **Standard Deviation:** This feature measures the dispersion of the signal around its mean value:

  $$\sigma_{e,t} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K} |c_{e,t}(k) - \bar{c}_{e,t}(k)|^2}, \tag{4}$$

  with $\bar{c}_{e,t}(k)$ the sample mean:

  $$\bar{c}_{e,t}(k) = \frac{1}{K}\sum_{k=1}^{K} c_{e,t}(k) \tag{5}$$

  and K the sample length of the excerpt considered.

- **Root Mean Square:** Square root of the averaged squared values of the signal excerpt:

  $$RMS_{e,t} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} |c_{e,t}(k)|^2} \tag{6}$$

- **Absolute Power Value:** Overall power was extracted from the PSD obtained, following previous research (Sammler et al., 2007)(Stancin et al., 2021), through Welch method, first proposed by (Welch, 1967).

  A Hamming window ($W(l), l = 0, ..., L-1$), with a 50% overlap was used for segmenting input signal into $S = 8$ segments of length L. The absolute power spectral value of the e-th electrode at the t-th trial is obtained by summing up the spectral estimation over all the frequency bins, $p(f_n)$, of the PSD obtained for the EEG excerpt considered, as follows:

  $$P_{e,t} = \sum_{n=0}^{L/2} p(f_n) \tag{7}$$

- **Averaged Spectral Flux:** Rate of change of the PSD of the input signal averaged over time. It is calculated by using:

  $$F_{e,t} = \frac{1}{S}\sum_{s=1}^{S} \sqrt{\sum_{n=0}^{L/2} |p(f_{n+1}) - p(f_n)|^2} \tag{8}$$

Following this approach, each trial outputs a feature matrix of dimensions (ExF), where $E = 61$ are the active electrodes used, and $F = 4$ are the features considered.

The aforementioned features have already shown compelling results for EEG signal characterization in musical mode detection (Guillén et al.). Note that all of them are energy-based, thus the process outlined can be viewed as a form of data augmentation. In contrast to typical EEG approaches where augmentation involves altering the dataset through noise addition or geometric transformations (Lashgari et al., 2020; George et al., 2022), the approach presented aims to optimize the information encompassed in the original dataset without altering it.

5 Results

This work proposes a feature-based model for the characterization and classification of EEG trials when processing "yes" and "no" words when presented in different communication channels such as text, symbol, sound, speech, or thought, with the final aim of helping the communication possibilities of people with some type of motor/neural disability that may diverge into communication difficulties. For the classification, a type of Long Short-Term Memory (LSTM) is used, reviewing both intra-, for each participant, and inter-subject, for all participants combined, scenarios. In (Rezazadeh Sereshkeh et al., 2017), a LSTM model was used for the decoding of "yes" and "no" as stated in section 1.

LSTM algorithms have shown their effectiveness in automatically predicting timeline properties (Algarni et al., 2022). Bi-LSTM classifier consists of an input sequence layer of $E = 61$ inputs, each input a vector built upon the features previously stated. Then, a bidirectional LSTM layer, built up of a forward layer and a backward layer, with 20 hidden units is used to learn the bidirectional long-term dependencies between sequence data flows. A fully-connected layer with two possible states ("yes" and "no" classes) is placed prior to outputting the label chosen for the data classified using a non-linear softmax layer supported by the cross entropy loss. A count of 1500 epochs is chosen to reach model convergence. Figure 3 shows the architecture of the model chosen.

This model is similar to the one used in (Ariza et al., 2022), though specific changes were done to adopt its structure to the task at hand, as described.

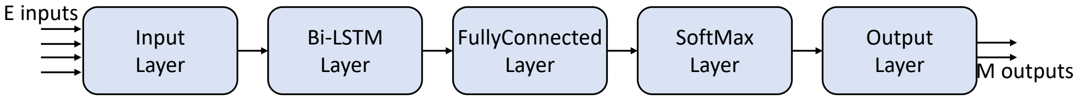


Figure 3: Diagram of the configuration of the Bi-LSTM network model used.

To avoid biases, and over-fitting in the classification step, and considering the dataset sample size, a 3-fold cross-validation process is carried out, so the results presented in further sections are the average of the 3 folds considered.

Using the data obtained after the experiment, matrices are labeled after the subject's response during each trial, depending on the concept processed: "yes" if the ongoing concept was "yes", or "no", otherwise. With this, binary classification was carried out. Table 1 shows Accuracy, Recall, Precision, and F1-score metrics obtained for each subject confusion matrix, along with their average value.

Table 1: Results (%) of binary classifications: "yes" or "no" word/concept processed.

Subject ID	Accuracy	Precision	Recall	F1-score
S$_1$	73.13	66.25	76.81	71.14
S$_2$	71.88	56.25	81.82	66.67
S$_3$	85.00	70.00	100.00	82.35
S$_4$	80.63	100.00	72.07	83.77
S$_5$	75.00	65.00	81.25	72.22
S$_6$	77.50	91.25	71.57	80.22
S$_7$	83.13	85.00	81.93	83.44
S$_8$	85.63	86.25	85.19	85.72
S$_9$	75.63	96.25	68.14	79.79
S$_{10}$	85.63	95.00	80.00	86.86
S$_{11}$	88.75	91.25	86.90	89.02
S$_{12}$	74.38	82.50	70.97	76.30
S$_{13}$	87.50	90.00	85.71	87.80
S$_{14}$	93.75	93.75	93.75	93.75
S$_{15}$	81.25	90.00	76.60	82.76
Averaged	81.25	83.92	80.85	81.45

In this table it can be observed that, although each subject outputs different results, the model's performance is consistent between them, and manages to surpass 81% of accuracy on average, proving this method to successfully discern when participants are processing one word or another regardless of the communication channel employed according to (Perelmouter and Birbaumer, 2000; Müller-Putz et al., 2008). These averaged values can be also observed in Figure 4, where the confusion matrix obtained by summing up all confusion matrices of subjects is presented. Note that the results attained are in line with (Choi and Kim, 2019) where a SVM model was used obtaining 86.03% of accuracy score, and surpass results from (Rezazadeh Sereshkeh et al., 2017) where a LSTM model was used, reaching up to 63.17%

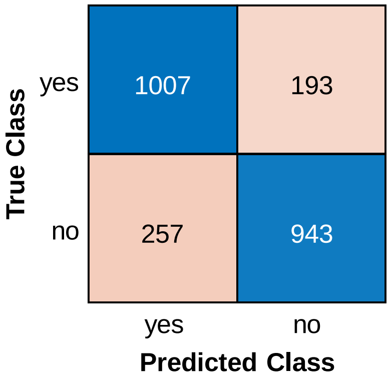


Figure 4: Average confusion matrix obtained by summing up all subject's confusion matrices.

AUC-ROC curves are drawn to support these metrics, which have shown to be directly correlated with the accuracy, but also considering the miss-classification cost and giving an indication of the amount of "work done" by the classification scheme evaluated (Bradley, 1997). Figure 5 shows the AUC-ROC curve obtained per user (colored and dotted), and the average curve obtained for all of them (black and continuous) in this scenario.

In this figure it can be observed how the model manages to convergence for all subjects, with an average AUC-ROC value of 92.29%, proving the model chosen to successfully discern between the words studied.

Inter-subject scenario was also assessed, but due to the experiment configuration, the model chosen did not manage to attain compelling results under the same training options chosen. This was somehow expected since in (Price, 2012) was stated that intra-operative stimulation showed diversity in location of language functions and morpho-metrical imaging studies based on diversity of brain shape and gyral patterns.

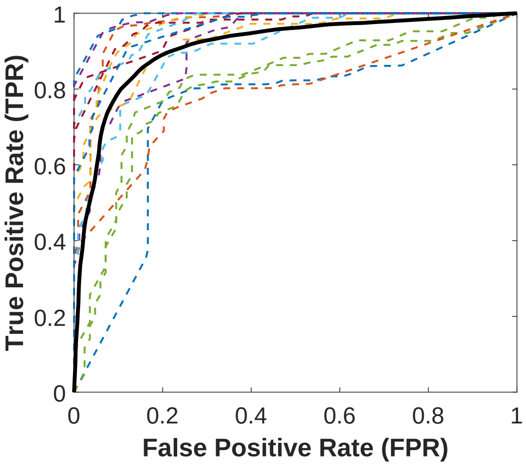


Figure 5: AUC-ROC curves comparison between all subjects curves (dotted-colored curves) and average curve (continuous-black curve) for binary classification using EEG data of all subjects when processing “yes” and “no” words/concepts.

6 Conclusions

In this paper, an analysis of brain responses to words/concepts with opposite meanings, i.e., “yes” and “no”, has been carried out using EEG signals. The final purpose of this work was to discern between them aiming to help the communication of people with some disability. For this purpose, the subjects participating in the experiment were presented the stimuli through several communication channels, i.e., text, symbol, sound, speech, and thought, trying to cover all the possible processing methods of the concepts. Note that all of them were considered since the control group was composed of healthy subjects. Nevertheless, it should be expected that when some kind of disability is present, at least one of them is still possible.

The scalp power distribution of all scenarios mentioned was reviewed, showing in all cases similar activation patterns, consistent with previous studies in language processing, which may indicate that both words/concepts are processed similarly regardless of the presentation medium. Based on this finding, all trials from different communication channels are used jointly in the processing and classification of signals.

Real cepstrum is used to characterize EEG signals once pre-processed. Then, energy-based features are extracted on a single-trial basis for each electrode individually.

Intra- and inter-subject scenarios are explored. The Bi-LSTM Neural Network model chosen successfully discerns between “yes” and “no” words/concepts regardless of the communication channel chosen in the former, attaining an average 81.25% accuracy value in the intra-subject binary classification scheme supported by an average AUC-ROC value of 92.29%, showing an improvement in the discrimination task when compared with previous researches. Inter-subject scenario was also assessed, but no compelling results were obtained maybe due to the variability of the multi-modal communication scheme considered and physiological differences across subjects.

Based on the results obtained, the processing scheme described in this work stands as a valuable tool to explore the possibility of enhancing the communication capabilities of people with some motor/neural disability by detecting simple words/concepts of opposite meaning. These findings encourage enlarging the dataset and continue the research.

Limitations

The main limitation of our study lies in the sample size at our disposal. This is a common limitation in EEG experiments where participants recruitment, specially without monetary compensation, is limited. However, although working with a reduced dataset might led to possible misinterpretations, sample size does not necessarily affect the validity of the research outcome, allowing the results obtained to be considered valid (Vozzi et al., 2021). Nevertheless, a larger sample could provide a more comprehensive and representative perspective. To overcome this drawback, it is expected to expand our dataset in further research stages.

Ethics Statement

The authors of this article declare that they have no known conflicts of interest and received no funding or financial support from any organization that could potentially bias the work reported in this paper.

Acknowledgements

This publication is part of project PID2021-123207NB-I00, funded by MCIN/AEI/10.13039/501100011033/FEDER, UE. This work was done at Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

References

Noor K. Al-Qazzaz, Reda J. Lafta, and Maimonah A. Khudhair. 2023. Estimations of emotional synchronization indices for brain regions using electroencephalogram signal analysis. *Advances in Non-Invasive Biomedical Signal Sensing and Processing with Machine Learning*, pages 315–344.

Mona Algarni, Faisal Saeed, Tawfik Al-Hadhrami, Fahad Ghabban, and Mohammed Al-Sarem. 2022. Deep learning-based approach for emotion recognition using electroencephalography (EEG) signals using bi-directional long short-term memory (bi-LSTM). *Sensors*, 22(8).

Mawadda Alghamdi, Saeed M. Qaisar, Shahad Bawazeer, Faya Saifuddin, and Majed Saeed. 2023. Brain-computer interface (BCI) based on the EEG signal decomposition butterfly optimization and machine learning. *Journal of medical signals and sensors*.

Electroencephalographic Society American. 1994. Guideline thirteen: Guidelines for standard electrode position nomenclature. *Journal of Clinic Neurophysiology*, 11(1):111–113.

Isaac Ariza, Lorenzo J. Tardón, Ana M. Barbancho, Irene De-Torres, and Isabel Barbancho. 2022. Bi-lstm neural network for EEG-based error detection in musicians' performance. *Biomedical Signal Processing and Control*, 78.

Jeffrey R. Binder, Rutvik H. Desai, William W. Graves, and Lisa L. Conant. 2009. Where Is the Semantic System? A Critical Review and Meta-Analysis of 120 Functional Neuroimaging Studies. *Cerebral Cortex*, 19(12):2767–2796.

Andrew P. Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

GmbH Brain Products. 2016. ActiCAP (64 channels) [apparatus]. Gilching, Germany. `https://www.brainproducts.com/solutions/acticap/`.

GmbH Brain Products. 2021. Brainvision recorder (version 2.2.2) [software]. Gilching, Germany. `https://www.brainproducts.com/solutions/recorder/`.

D.G. Childers, D.P. Skinner, and R.C. Kemerait. 1977. The cepstrum: A guide to processing. *Proceedings of the IEEE*, 65(10):1428–1443.

Jeong Woo Choi and Kyung Hwan Kim. 2019. Covert intention to answer "yes" or "no" can be decoded from single-trial electroencephalograms (EEGs). *Computational Intelligence and Neuroscience*, (1):4259369.

Mike X Cohen. 2014. *Analyzing Neural Time Series Data*, 1st edition. The MIT Press.

Mahmood Danyal, Riaz H. Naseem, and Nisar Humaira. 2023. Introduction to non-invasive biomedical signals for healthcare. *Advances in Non-Invasive Biomedical Signal Sensing and Processing with Machine Learning*, pages 1–24.

Olawunmi George, Roger Smith, Praveen Madiraju, Nasim Yahyasoltani, and Sheikh I. Ahamed. 2022. Data augmentation strategies for EEG-based motor imagery decoding. *Heliyon*, 8(8).

Mick Grierson and Chris Kiefer. 2014. Contemporary approaches to music BCI using P300 Event Related Potentials. *Guide to Brain-Computer Music Interfacing*, pages 43–59.

Sergio Guillén, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Neural processing of musical mode through EEG signals. *Engineering Applications of Artificial Intelligence*. (under review).

John E. Hall and Michael E. Hall. 2021. *Textbook of Medical Physiology*, 14th edition. Elsevier.

Siqi Han, Chao Zhang, Jiaxin Lei, Qingquan Han, Yuhui Du, Anhe Wang, Shuo Bai, and Milin Zhang. 2024. Cepstral analysis-based artifact detection, recognition, and removal for prefrontal EEG. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(2):942–946.

Fatima Hassan and Syed F. Hussain. 2023. Review of eeg signals classification using machine learning and deep-learning techniques. *Advances in Non-Invasive Biomedical Signal Sensing and Processing with Machine Learning*, pages 159–183.

Gregory Hickok and David Poeppel. 2007. The cortical organization of speech processing. *Nature reviews. Neuroscience*, 8:393–402.

George H. Klem, Hans Lüders, Herbert H. Jasper, and Christian Erich Elger. 1999. The ten-twenty electrode system of the international federation. the international federation of clinical neurophysiology. *Electroencephalography and clinical neurophysiology. Supplement*, 52:3–6.

Elnaz Lashgari, Dehua Liang, and Uri Maozv. 2020. Data augmentation for deep-learning-based electroencephalography. *Journal of Neuroscience Methods*, 346.

Iolietta Lazarou, Spiros Nikolopoulos, Panagiotis C. Petrantonakis, Ioannis Kompatsiaris, and Magda Tsolaki. 2018. EEG-based brain–computer interfaces for communication and rehabilitation of people with motor impairment: A novel approach of the 21st century. *Frontiers in Human Neuroscience*, 12(14).

Steven J. Luck. 2014. *An Introduction to the Event-Related Potential Technique*, 2nd edition. The MIT Press. ISBN: 978-0-262-52585-5.

Inc. MathWorks. 2022. MATLAB version: 9.12.0.2039608 (r2022a update 5) [software]. Natick, Massachusetts, United States. `https://www.mathworks.com/`.

Gernot Müller-Putz, Reinhold Scherer, Clemens Brunner, Robert Leeb, and Gert Pfurtscheller. 2008. Better than random? a closer look on bci results. *International Journal of Bioelektromagnetism*, 10:52–55.

El Imrani Nizara. 2018. Una revisión de la neuroanatomía y neurofisiología del lenguaje. *Revista Neuro-Psiquiatría*, 8(3):196–202.

Robert Oostenveld, Pascal Fries, Eric Maris, and Jan Mathijs Schoffelen. 2010. Fieldtrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, 2011:9.

Aniruddh D. Patel. 2007. *Music, Language and the Brain*, 1st edition. Oxford University Press USA.

J Perelmouter and N Birbaumer. 2000. A binary spelling interface with random errors. *IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 8(2):227—232.

Steve Petersen, P Fox, Michael Posner, Mintun MA, and Marus Raichle. 1988. Positron emission tomographic studies of the cortical anatomy of single-word processing. *Nature*, 331:585–9.

Cathy J. Price. 2012. A review and synthesis of the first 20years of PET and fMRI studies of heard speech, spoken language and reading. *NeuroImage*, 62(2):816–847.

Alborz Rezazadeh Sereshkeh, Robert Trott, Aurélien Bricout, and Tom Chau. 2017. EEG classification of covert speech using regularized neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2292–2300.

Daniela Sammler, Maren Grigutsch, Thomas Fritz, and Stefan Koelsch. 2007. Music and emotion: Electrophysiological correlatees of the processing of pleasant and unpleasant music. *Psychophysiology*, pages 293–304.

Saeid Sanei and Jonathon A Chambers. 2013. *EEG signal processing*. John Wiley & Sons.

Ovishake Sen, Anna M. , Pranay R. Raman, Kabir S. Khara, Adam Khalifa, and Baibhab Chatterjee. 2023. Machine-learning methods for speech and handwriting detection using neural signals: A review. *Sensors*, 23(12):5575.

Nasrin Shourie. 2016. Cepstral analysis of EEG during visual perception and mental imagery reveals the influence of artistic expertise. *Journal of medical signals and sensors*, 6(4).

I. Stancin, M. Cifrek, and A. Jovic. 2021. A review of EEG signal features and their application in driver drowsiness detection systems. *Sensors*, 21.

Alejandro Villena, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho, Elvira Brattico, and Niels T. Haumann. 2019. Preprocessing for lessening the influence of eye artifacts in EEG analysis. *Applied Sciences*, 9(9).

Alessia Vozzi, Vincenzo Ronca, Pietro Aricò, Gianluca Borghini, Nicolina Sciaraffa, Patrizia Cherubino, Arianna Trettel, Fabio Babiloni, and Gianluca Di Flumeri. 2021. The sample size matters: To what extent the participant reduction affects the outcomes of a neuroscientific research. a case-study in neuromarketing field. *Sensors*, 21(18).

A. Eschman W. Schneider and A. Zuccolotto. 2016. E-prime 3.0 user's guide [software]. Pittsburgh: Psychology Software Tools, Inc.. `https://pstnet.com/products/e-prime/`.

P. Welch. 1967. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73.

Modeling Score Estimation for Japanese Essays with Generative Pre-trained Transformers

Boago Okgetheng and **Koichi Takeuchi**
Graduate School of Environmental, Life, Natural Science and Technology
Okayama University, Japan
`pcqm1k3t@s.okayama-u.ac.jp` `takeuc-k@okayama-u.ac.jp`

Abstract

This paper presents a study on Japanese essay grading using Generative Pre-trained Transformers (GPTs) in Japanese language. Previous research has demonstrated the effectiveness of neural network-based models, such as BERT, for essay grading across various datasets. With the advent of downloadable GPT models trained on significantly larger datasets compared to BERT, it has become feasible to employ these models for essay grading through fine-tuning with Low-Rank Adaptation (LoRA). Most existing models have focused on English essays and their accuracy, leaving a gap in understanding the performance on Japanese essays, which have limited linguistic resources. To address this, we apply several Japanese GPT models to a dataset comprising 12 prompts across 4 themes. The experimental results show that the model pre-trained exclusively on Japanese data, open-calm-medium, achieved an accuracy of 62.33% and a QWK of 0.5551. In comparison, the best-performing model additionally pre-trained on multilingual Llama, ELYZA-Llama-2-7b-fast, achieved an accuracy of 53.29% and a QWK of 0.3375. This study highlights the potential of GPT models for enhancing automated essay scoring in the Japanese context.

1 Introduction

Automated essay scoring (AES) is one of the most promising and rapidly evolving fields in educational technology owing to the growing opportunities of online lectures.

Previous studies first revealed neural network-based models such as LSTM and CNN are effective for essay tasks (Taghipour and Ng, 2016; Dong et al., 2017; Yi Tay and Minh C. Phan and Luu Anh Tuan and Siu Cheung Hui, 2018). A neural network-based essay scoring model is roughly divided into two parts: encoding an essay to a vector and assigning scores. After a pre-trained language model BERT (Devlin et al., 2019) has succeeded in improving the accuracy of benchmarks in NLP, some previous studies have applied simple BERT-based models into essay scoring task (Rodriguez et al., 2019; Mayfield and Black, 2020). The simple models were unable to improve the accuracy of existing neural network-based models. The newly proposed models, however, combining regression and ranking loss show improved performance comparing to the existing neural network-based models (Yang et al., 2020; Wang et al., 2022).

Thus, the previous studies have revealed pre-trained language models are effective for AES. In the recent advancements in Generative Pre-trained Transformers (GPTs) (Brown et al., 2020; OpenAI et al., 2023), which have much larger weight size and are trained on extensive datasets, several studies have explored the application of GPTs, both with and without fine-tuning (Mizumoto and Eguchi, 2023; Xiao et al., 2024). It has been observed that a prompt-based GPT model yields lower accuracy compared to the fine-tuned GPT-3.5 or BERT-based model (Xiao et al., 2024).

The findings of the models studied above have been often conducted on the commonly used English essay dataset ASAP (Hamner et al., 2012), but on the other hand, it is not clear how much prediction accuracy can be achieved for Japanese essays, where linguistic resources are limited. There are studies conducted on Japanese essay written by Japanese learners (Hirao et al., 2020; Obata et al., 2023); however, Japanese essay data (Takeuchi et al., 2021)[1] written by native Japanese speakers that can be used for research has recently been published, thus, in this paper, we conduct on the study of essay scoring model for Japanese.

Previous studies show that the fine-tuned language models based on BERT or GPT-3.5 are promising for AES task (Hirao et al., 2020; Xiao

---

[1] GSK2021-B https://www.gsk.or.jp/catalog/gsk2021-b/

et al., 2024). Thus, the middle size of downloadable GPT models such as Llama (Touvron et al., 2023) are worth to be applied into Japanese essay scoring task because of the following reasons: 1) API-based GPTs such as GPT-3.5 have limitations of learning while we can freely build an essay grading model that incorporate the downloaded GPT, 2) it is expected that linguistic knowledge within a GPT will contribute to solve the grading of Japanese essays, and 3) Low-Rank Adaptation (LoRA) (Hu et al., 2021) enables us to apply fine-tuning on a local GPU at a laboratory scale.

Several Japanese GPT models that are specifically pre-trained on Japanese texts are published; however, it is not clear which model is suitable for Japanese essay scoring task. The dataset includes Japanese essays to 12 prompts consists of 4 themes, which ranges in length from 100 to 800 characters. Therefore, in this paper, we clarify the performance of the several Japanese GPT models for the Japanese essay dataset and discuss the relations between GPTs and features of essays.

The contributions of this study are as follows: 1) it unveils Quadratic Weighted Kappa (QWK) and F1 scores achieved for Japanese essays using a Japanese GPT model, 2) it provides a comparative analysis of the performance across various Japanese GPT models employing Low-Rank Adaptation (LoRA) fine-tuning on Japanese essay datasets, and 3) it reveals that GPT models initially trained on Japanese texts outperform the model subjected to additional pre-training on multilingual Llama model using Japanese texts.

2 Previous Studies

In the initial phases of AES development, a variety of statistical models were employed. These included regression models that relied on handcrafted features, exemplified by systems like e-rater (Attali and Burstein, 2006), as well as statistical approaches utilizing latent semantic indexing (LSI) (Deerwester et al., 1990; Ishioka and Kameda, 2006).

Neural network models that do not require handcrafted features has been proposed and shown to be superior to previous models. Many studies used LSTM and CNN models (Taghipour and Ng, 2016; Dong et al., 2017; Yi Tay and Minh C. Phan and Luu Anh Tuan and Siu Cheung Hui, 2018), but there is also a study using word embedding and Support Vector Regression model (Cozma et al., 2018) that achieved an equivalent performance to the neural network-based models (Mayfield and Black, 2020).

Instead of learning sentence embedding directly from target data, pre-trained language models are employed (Rodriguez et al., 2019; Mayfield and Black, 2020; Yang et al., 2020; Wang et al., 2022; Mizumoto and Eguchi, 2023; Xiao et al., 2024; Hirao et al., 2020; Obata et al., 2023). Pre-trained models can be broadly divided into BERT (Rodriguez et al., 2019; Mayfield and Black, 2020; Yang et al., 2020; Hirao et al., 2020; Wang et al., 2022) and GPT (Mizumoto and Eguchi, 2023; Obata et al., 2023; Xiao et al., 2024). Although the initial model using BERT could not achieve high accuracy, it was shown that adding ranking to the loss function improved accuracy and outperformed neural network-based models (Yang et al., 2020; Wang et al., 2022). The prompt-based GPT model showed the limited performance compared to the linguistic feature-based model (Mizumoto and Eguchi, 2023; Obata et al., 2023) or fine-tuned GPT-3.5 model (Xiao et al., 2024). This indicates that significant large language model is not so effective for AES.

While most of the previous studies are conducted on English essay dataset, studies on Japanese essay are limited. Hirao et al. (2020) revealed that the BERT-based model is effective compared to the LSTM-based model on Japanese essay dataset[2]. The other Japanese essay dataset used in Obata et al. (2023) contains essays for one prompt[3]. Preliminary experiments have been conducted to predict scores for Japanese essay data by fine-tuning Japanese GPT models (Okgetheng and Takeuchi, 2024).

Thus, evaluating essay scoring models using a Japanese essay dataset—comprising essays of various lengths and themes, based on data available for research—is deemed valuable.

3 Methodology

3.1 Essay Scoring Model

The essay scoring model comprises two main modules: text encoding and score assignment. The encoding module leverages pre-trained language models to convert the input text into vector representations, while the score assignment module

---

[2] https://goodwriting.jp/wp/?lang=en

[3] That is included in I-JAS corpus https://www2.ninjal.ac.jp/jll/lsaj/.

utilizes these representations to predict scores. The models employed in this study include Japanese BERT, Open CALM, CALM2-7B, StableLM Alpha, and ELYZA, each designed specifically for handling Japanese texts.

Japanese BERT[4] is used for text encoding, where the vector corresponding to the [CLS] token serves as the embedding vector for the input essay. In contrast, decoder-only models such as Open CALM[5], CALM2-7B[6], Japanese StableLM Alpha[7], and ELYZA[8] are utilized for both encoding and score prediction. For these GPT-based models, the vector that predicts the next token after the final token of the input essay is used as the embedding vector.

Given an input essay document s with tokens x_1 to x_n generated by the tokenizer, the final token embedding is used for predicting the score. Specifically, for models like Open CALM, the vector corresponding to the token that denotes the end of the input document is used. Figure 1 illustrates the overall architecture of the essay scoring model.

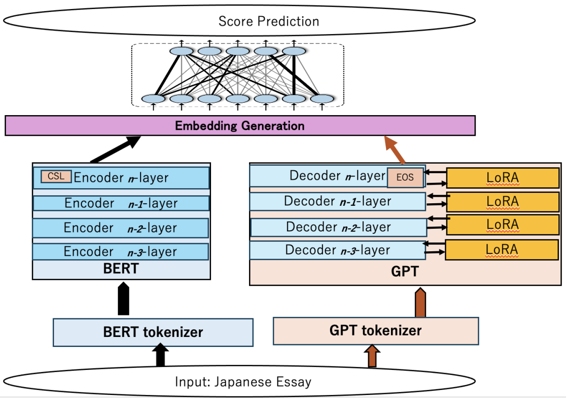


Figure 1: Methodology for the Neural Network-based Essay Scoring Model

3.2 Score Prediction from Embeddings

To predict the score from the embeddings, the final embedding vector (obtained either from the [CLS] token for BERT or the end-of-sequence token for GPT models) is passed through a fully connected neural network. This network consists of multiple layers that map the high-dimensional embeddings to a single score value representing the predicted essay score. The design of this neural network, including the number of layers and activation functions, is optimized to capture the nuanced relationships between the encoded text and the target scores.

3.3 Design of the Loss Function

Given that the proposed model is a categorical classification model where the classes are ordinal, we applied soft labeling(Diaz and Marathe, 2019) to the loss function. During the training phase, the loss for the categorical model is calculated using cross-entropy with one-hot labels. Soft labeling modifies the target labels such that the k-th value is calculated as follows:

$$d_k = \frac{exp(-|\hat{k} - k|)}{\sum_{i=1}^{K} exp(-|\hat{k} - i|)} \tag{1}$$

Here, d_k represents the teacher value for each k-th unit in the final layer of the classification model, and $\hat{k}$ denotes the correct category. This approach assigns a larger penalty for predictions that are further from the correct answer, promoting better ordinal classification.

4 Experimental Setup

4.1 Dataset

The Japanese essay tests were conducted on Japanese university students, and the dataset consists of 12 prompts with 4 themes. In each theme, there are three prompts. The four themes are globalization (Global), natural science (Natural), East Asian economics (Easia), and critical thinking (Criticize). Each theme has three prompts from question 1 to 3. The length of the essays ranges from 100 characters to 800 characters.

The essays are manually scored on a 5-point scale for comprehension, logic, validity, and grammar. In this paper, we focus on comprehension scores to evaluate the essay scoring models. The essays were annotated by two Japanese-speaking raters, and the scores were averaged to obtain the final score for each essay.

The Japanese essay data is available to researchers and is provided by the Japanese Language Resource Association (GSK)[9]. Table 1 shows the number of essays for each prompt. In the table, ’P’ stands for Prompt number, ’ML’ represents the Maximum Length of an essay, and ’Num’ indicates the number of essays.

---

[4] https://huggingface.co/tohoku-nlp/bert-base-japanese-v3

[5] https://huggingface.co/cyberagent/open-calm

[6] https://huggingface.co/cyberagent/calm2-7b

[7] https://huggingface.co/stabilityai/japanese-stablelm-base-alpha-7b

[8] https://huggingface.co/elyza/ELYZA-japanese-Llama-2-7b

[9] https://www.gsk.or.jp/en/

This dataset provides a diverse range of essay lengths and topics, enabling a comprehensive evaluation of the essay scoring models.

Table 1: Japanese essay data

Theme	P	ML	Num	Theme	P	ML	Num
	1	100	290		1	300	328
Criticize	2	400	290	Global	2	250	327
	3	800	290		3	300	327
	1	300	290		1	100	327
Easia	2	250	288	Science	2	400	325
	3	300	288		3	800	327

4.1.1 Example of Global Category Prompts: Japanese and English Versions

In the global category, the essay prompts challenge students to critically analyze various aspects of globalization. For example, Prompt 1 asks: **Japanese:** グローバリゼーションは、世界、または各国の所得格差をどのように変化させましたか。また、なぜ所得格差拡大、または縮小の現象が現れたと考えますか。300字以内で答えなさい。 **English:** How has globalization changed income inequality in the world or across countries? Also, why do you think the phenomenon of increasing or decreasing income inequality has appeared? Please answer within 300 characters.

Prompt 2 shifts focus to multinational corporations, asking: **Japanese:** 多国籍企業は、グローバリゼーションの進展の中でどのような役割を果たしましたか。多国籍業の具体例をあげて、250字以内で答えなさい。 **English:** What role have multinational corporations played in the development of globalization? Give a specific example of a multinational business and answer within 250 characters.

Lastly, Prompt 3 delves into cultural aspects, asking: **Japanese:** 文化のグローバリゼーションは、私たちの生活にどうのような影響を与えましたか。また、あなたはそれをどのように評価しますか。具体例をあげて、300字以内で答えなさい。 **English:** How has cultural globalization affected our lives? Also, how do you rate it? Give a specific example and answer within 300 characters.

4.2 Score Distribution Across Themes

The score distribution across different essay themes and prompts provides valuable insights into the grading trends and the level of challenge posed by each prompt. Figure 2 illustrates how scores were allocated across five possible score levels (1 to 5) for each theme and prompt within the dataset. This

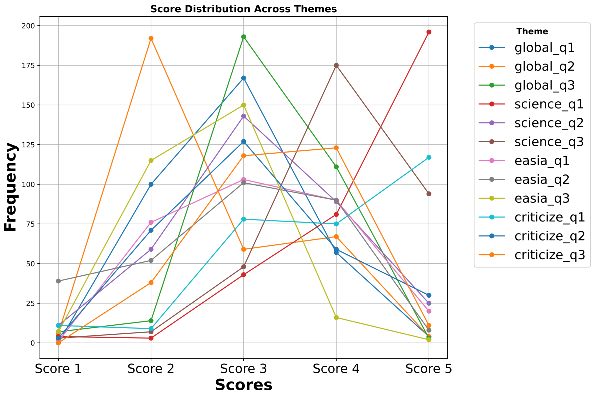


Figure 2: Scores Distribution per theme

distribution highlights the variability in grading across different prompts, with some prompts showing a higher concentration of scores in the middle ranges (Scores 2 and 3), while others have a significant number of essays scored at the higher end (Score 5), particularly in themes like **science_q1.**

4.3 Performance Measures

To evaluate the effectiveness of our model, we employed several performance metrics:

- Accuracy: This metric provided a straightforward measure of the model's ability to correctly predict the essay scores.
- Root Mean Square Error (RMSE): RMSE offered a quantitative measure of the model's prediction error, giving insights into the deviation of the predicted scores from the actual scores.
- Quadratic Weighted Kappa (QWK): QWK was used to assess the degree of agreement between the predicted and actual essay scores. This metric is particularly valuable in grading scenarios, as it accounts for the ordered nature of the rating scale.

4.4 Training Setup

Our setup involved the following key components:

- GPT Configuration: We utilized GPT models specifically configured for the Japanese language, ensuring that they are finely attuned to the linguistic characteristics unique to Japanese.

- Early Stopping: To prevent overfitting, we employed an early stopping mechanism. Training ceased once the improvement in performance on the validation set plateaued, ensuring the generalizability of the model.
- Gradient Accumulation: Recognizing the computational demands of training large language models, we implemented a gradient accumulation strategy. By setting the accumulation steps to 2 with a batch size of 8, we effectively simulated a larger batch size of 16, allowing for more stable and effective training.
- LoRA: We applied LoRA (Low-Rank Adaptation) implemented in PEFT (Parameter-Efficient Fine-Tuning) by HuggingFace with the rank set to 8.
- Training Configuration: Models were trained over a maximum of 10 epochs with early stopping criteria to prevent overfitting.

5 Experimental Results

In our experiments, we employed a 5-fold cross-validation technique to ensure the robustness and reliability of our results. Each model was trained with a batch size of 8, and we used a gradient accumulation step of 2, effectively making the batch size 16. The models were trained for a maximum of 10 epochs, with early stopping criteria to prevent overfitting.

The performance metrics used in our evaluation include F1 Score, QWK, Accuracy, and RMSE. These metrics provide a comprehensive evaluation of the models' capabilities in handling classification tasks, measuring the agreement between predicted and actual scores, assessing the proportion of correct predictions, and quantifying the average magnitude of prediction errors, respectively.

5.1 Overall Performance

Table 2 presents the overall performance of various models with and without soft labeling.

This table shows that models such as calm2-7b and open-calm-large perform consistently well across all metrics. Specifically, calm2-7b without soft labeling achieves the highest QWK (0.5982) and a relatively low RMSE (0.6957), indicating strong agreement with the true scores and precise predictions. In contrast, the F1 scores are generally higher for models without soft labeling, suggesting a better precision-recall balance when soft labels are not used.

5.2 Category-wise Performance

Table 3 illustrates the performance of different models across various essay categories with and without soft labeling. The results in this table are for the models that performed best in each category.

In the Criticize category, the calm2-7b model without soft labeling outperforms other models, achieving a QWK of 0.5831 and RMSE of 0.7133. The Easia category shows similar trends, with calm2-7b again performing best without soft labeling. For the Science category, the open-calm-medium model with soft labeling achieves the highest QWK of 0.7092, indicating strong performance in more technical essays.

5.3 Prompt-wise Performance

Table 4 provides the performance across different prompts with and without soft labeling. In this table, we are showing the results of the models that performed better than the others in each prompt.

For Prompt 1, the jp(Japanese)-stablelm-instruct-7b-v2 model without soft labeling achieves the highest QWK of 0.7356, indicating a strong agreement with human scoring. Prompt 2 shows the ELYZA-Llama-2-7b-fast-instruct model performing well, with balanced accuracy and F1 score. The calm2-7b model remains consistent across different prompts, showcasing its versatility.

5.4 Performance Comparison

Table 5 compares the performance of classification models with soft labeling, without soft labeling, and regression models.

Table indicates that regression models generally outperform classification models in terms of RMSE, indicating more precise error minimization. Soft labeling improves performance for medium and large models, but its benefits are less clear for small models. QWK and Accuracy metrics show balanced performance across all model types, with regression models slightly ahead in precision.

6 Discussions

The analysis of various models on the Japanese essay scoring task demonstrates that some models exhibit a high degree of proficiency within certain thematic areas. This is evidenced by their consistently strong performance across most evaluated

Table 2: Overall Performance of GPT Models

	Model	F1 Score	QWK	Accuracy	RMSE
With Soft Labeling	open-calm-small	0.2803	0.3417	0.5677	0.7855
	open-calm-medium	0.3284	**0.5303**	0.5899	**0.7243**
	open-calm-large	0.3502	0.5272	**0.6208**	0.7282
	open-calm-7b	0.3072	0.4362	0.5963	0.7787
	calm2-7b	0.3252	0.5288	0.6001	0.7417
	calm2-7b-chat	0.3109	0.4512	0.5873	0.7761
	jp-stablelm-alpha-7b	0.2961	0.4201	0.5652	0.7933
	jp-stablelm-instruct-7b-v2	0.3372	0.4750	0.5886	0.7788
	ELYZA-Llama-2-7b-instruct	0.2909	0.3760	0.5305	0.8980
	ELYZA-Llama-2-7b-fast	0.2415	0.3105	0.5216	0.8884
	ELYZA-Llama-2-7b	0.3372	0.4716	0.5930	0.7728
	ELYZA-Llama-2-7b-fast-instruct	0.3115	0.4376	0.5481	0.7893
	BERT	**0.5056**	0.4318	0.5602	0.7863
Without Soft Labeling	open-calm-small	0.2910	0.3848	0.5679	0.8112
	open-calm-medium	0.3621	0.5551	**0.6233**	0.7259
	open-calm-large	0.3772	0.5614	0.6219	0.7053
	open-calm-7b	0.3370	0.5068	0.6089	0.7279
	calm2-7b	0.3872	**0.5982**	0.6140	**0.6957**
	calm2-7b-chat	0.3303	0.4994	0.6072	0.7332
	jp-stablelm-alpha-7b	0.3518	0.5367	0.6072	0.7332
	jp-stablelm-instruct-7b-v2	0.3362	0.4690	0.5918	0.7829
	ELYZA-Llama-2-7b-instruct	0.3143	0.4501	0.5274	0.8365
	ELYZA-Llama-2-7b-fast	0.2630	0.3375	0.5329	0.9217
	ELYZA-Llama-2-7b	0.3526	0.4843	0.5768	0.8207
	ELYZA-Llama-2-7b-fast-instruct	0.3260	0.4495	0.5520	0.8053
	BERT	**0.4681**	0.3352	0.5450	0.8433

Table 3: Category-wise Performance of GPT Models

	Category	Model	QWK	RMSE	Accuracy	F1 Score
With Soft Labeling	Criticize	jp-stablelm-instruct-7b-v2	0.5239	0.7287	0.6061	0.3395
	Easia	calm2-7b	0.5129	0.6259	0.6919	0.3119
	Global	open-calm-large	0.5593	0.7810	0.5690	0.3857
	Science	open-calm-medium	0.7092	0.6604	0.6667	0.4515
Without soft labeling	Criticize	calm2-7b	0.5831	0.7133	0.5960	0.3805
	Easia	calm2-7b	0.5886	0.6280	0.6818	0.3620
	Global	calm2-7b-chat	0.5585	0.6511	0.6149	0.4092
	Science	jp-stablelm-alpha-7b	0.7050	0.6565	0.6061	0.4277

Table 4: Prompt-wise Performance of GPT Models

	Prompt	Model	QWK	RMSE	Accuracy	F1 Score
With Soft Labeling	1	jp-stablelm-instruct-7b-v2	0.6881	0.6541	0.6869	0.4352
	2	calm2-7b-chat	0.6963	0.7388	0.5606	0.3603
	3	open-calm-large	0.4243	0.7100	0.6300	0.3082
Without Soft Labeling	1	jp-stablelm-instruct-7b-v2	0.7356	0.6070	0.7355	0.4835
	2	ELYZA-Llama-2-7b-fast-instruct	0.6920	0.6931	0.5990	0.3932
	3	calm2-7b	0.4373	0.6922	0.5917	0.3440

Table 5: Performance Comparison using Classification Model with Soft Labeling (WS), Without Soft Labeling (WOS) and Regression Model (RM)

	Small			Medium			Large		
Metric	WS	WOS	RM	WS	WOS	RM	WS	WOS	RM
F1 Score	0.2803	0.2910	**0.5109**	0.3284	0.3621	**0.5552**	0.3502	0.3772	**0.5358**
QWK	0.3417	0.3848	**0.3872**	0.5303	**0.5551**	0.4521	0.5272	**0.5614**	0.3528
Accuracy	0.5677	**0.5679**	0.5441	0.5899	**0.6233**	0.5980	0.6208	**0.6219**	0.5882
RMSE	0.7855	0.8112	**0.6826**	0.7243	0.7259	**0.6511**	0.7282	0.7053	**0.6793**

metrics. Such results suggest that these models do better on predicting scores in that thematic area.

While BERT's performance was not the strongest, it did achieve commendable results in the F1 measure across all themes, indicating a balanced precision and recall in the classification task. However, in comparison to GPT models, BERT was surpassed in other key metrics, suggesting that while BERT is proficient in identifying relevant instances, GPT models may offer a more comprehensive understanding of the dataset, reflecting a deeper contextual grasp that extends beyond mere classification accuracy.

The analysis of prompt lengths in relation to essay difficulty reveals that longer prompts, such as Criticize prompt 3 and Science prompt 3, do not necessarily correlate with increased challenge levels. Contrastingly, Prompt 2 stands out, where despite its shorter length, human graders scored it as more difficult, indicating that the inherent complexity of a prompt and the resultant essay responses are not solely determined by length. This insight suggests that prompt difficulty could be influenced by the intricacy of the topic and the cognitive demands it places on the essay writers.

The research sought to gain deeper insights into the effectiveness of using a Regression Model (RM) for classification tasks and results were recorded in Table 5 for 3 GPT models (calm small, medium and large). In the Japanese essay scoring task, it was found that models employing the classification model with soft labeling (WS) generally had superior performance in terms of QWK compared to those using the classification model without soft labeling (WOS) and the regression model . This suggests that soft labeling models are better at accounting for the ordinal nature of the grading task. Although the regression models using Mean Square Error loss achieved the highest F1 Scores, this did not consistently extend to higher accuracy or QWK. Such findings indicate that while RM is proficient at minimizing the variance of the errors, it may not always translate into the most accurate categorization, especially when the task requires understanding the ordered grading system.

When evaluating the differences in the pre-training methods among the models in Table 2, the GPT models trained on Japanese texts from the beginning (i.e., open-calm, calm2-7b and jp-stable models) outperform the model subjected to continual pre-training on multilingual Llama model (i.e., ELYZA) for Japanese texts. Since there is only one model of continuous pre-trained model, however, this outcome presents intriguing prospects for future insights into pre-trained models.

7 Conclusions

In this paper, we have expanded the AES field by applying GPTs to Japanese essay grading—a linguistic domain previously underexplored due to limited resources. Our research demonstrates that Japanese-specific pre-trained GPT models, particularly when fine-tuned with LoRA, can effectively navigate the complex linguistic landscape of Japanese and provide accurate essay assessments. The research revealed that models pre-trained exclusively on Japanese corpora outperformed their counterparts fine-tuned from multilingual datasets, highlighting the importance of tailored linguistic training in automated essay scoring systems.

The calm2-7b model demonstrated exceptional capability, consistently achieving high scores across various evaluation metrics, including QWK and RMSE especially in Easia theme. Its robust performance across this topic underscores its suitability as a precise and reliable tool for the automated grading of Japanese essays in this thematic area.

This study not only contributes a significant finding to the field of educational technology but also opens avenues for the deployment of language-specific automated grading tools.

8 Limitations

The study faced limitations in data availability, model architecture, and computational resources, particularly GPU memory constraints, which may have impacted the training efficiency and model performance.

9 Ethical Considerations

Ethical considerations were rigorously adhered to, ensuring the protection of individual privacy. The dataset did not contain any personal information, guaranteeing the anonymity of all individuals involved. The data employed is publicly available, reinforcing the ethical integrity of our research.

Acknowledgements

Part of this study was supported by JSPS KAKENHI Grandt Number JP22K00530.

References

Yigal Attali and Jill Burstein. 2006. Automated Essay Scoring with e-rater V.2. *The Journal of Technology, Learning, and Assessment*, 4(3):1–30.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. arXiv:2005.14165.

Madalina Cozma, Andrei M. Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *R.E. Asher (Editor-in-Chief), The Encyclopedia of Language and Linguistics, Vol.6, Oxford: Pergamon Press*, pages 3168–3171.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(7):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4738–4746.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring.

Reo Hirao, Mio Arai, Hiroki Shimanaka, Satoru Katsumata, and Mamoru Komachi. 2020. Automated essay scoring system for nonnative Japanese learners. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1250–1257, Marseille, France. European Language Resources Association.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv:2106.09685.

Tsunenori Ishioka and Masayuki Kameda. 2006. Automated Japanese Essay Scoring System based on Articles Written by Experts. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL.*

Elijah Mayfield and Alan W Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics.

Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2).

Ayaka Obata, Takumi Tagawa, and Yuichi Ono. 2023. Assessment of ChatGPT's validity in scoring essays by foreign language learners of japanese and english. In *Proceeding of the 15th International Congress on Advanced Applied Informatics*.

Boago Okgetheng and Koichi Takeuchi. 2024. Estimating japanese essay grading scores with large language models. In *Proceedings of the 30th Annual Conference on Natural Language Processing (NLP)*, pages 643–647, Japan. NLP Society. This work is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. arXiv:2303.08774.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and automated essay scoring. arXiv:1901.07744.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Koichi Takeuchi, Masayuki Ohno, Kouta Motojin, Masahiro Taguchi, Yoshihiko Inada, Masaya Iizuka, Tatsuhiko Abo, and Hitoshi Ueda. 2021. Development of Essay Scoring Methods Based on Reference Texts with Construction of Research-Available Japanese Essay Data. In *IPSJ Journal Vol.62 No.9*, pages 1586–1604. (in Japanese).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Changrong Xiao, Wenxing Ma, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2024. From automation to augmentation: Large language models elevating essay scoring landscape. arXiv:2401.06431.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Yi Tay and Minh C. Phan and Luu Anh Tuan and Siu Cheung Hui. 2018. SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text. In *Proceedings of the Thirty-Second AAAI Conference on Artifical Intelligence*, pages 5948–5955.

CliqueCorex: A Self-supervised Clique-based Anchored Topic Model

Sami Diaf
Universität Hamburg
Department of Socioeconomics
`sami.diaf@uni-hamburg.de`

Abstract

Probabilistic generative topic models are the de facto choice for most text data applications, usually augmented with unsupervised and semi-supervised learning strategies to enhance the topic quality. Alternatively, information theory was used to build model-free algorithms able to learn homogeneous, binary latent groups of words, as topics, via multivariate mutual information as for the *Correlation Explanation* model (*CorEx*), with the possibility of incorporating anchors, or keywords, as prior information that better reflects the practitioner's experience to reveal nested topics. This paper establishes a self-supervised, anchor-based strategy, namely *CliqueCorex*, where anchors are meaningful subgraphs resulting from the hierarchical clustering of the corpus' bigrams via *clique percolation* algorithm. This scheme maximizes the information extraction by learning cohesive topics without biased prior information or any additional hyperparameter optimization. Applied to two central banking corpora, *CliqueCorex* improved the plain *CorEx* results without the need of additional topics, while uncovering nested topic contents, spanning across a wide spectrum of monetary policy practices, with a natural separability and an importance order that demonstrate the usefulness of cliques when implementing a guided inference.

1 Introduction

The abundance of textual sources and their growing complexity has led to continuous attempts to improve the existing text-as-data methods. These efforts have either sought sophistication from neural networks, or improved existing generative models to better handle the studied task (Churchill and Singh, 2022), at the expense of detailed hyperparameter specifications (Airoldi et al., 2014; Gallagher et al., 2017).

In machine learning, probabilistic topic models are still considered as the workhorse for most text mining applications, particularly the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) whose scheme has been adopted later by many strategies that brought improvements via adding metadata as covariates (Blei and McAuliffe, 2007), time-based topics (Blei and Lafferty, 2006) or nested hierarchies (Griffiths et al., 2003). Semi-supervised approaches (Lu et al., 2011; Jagarlamudi et al., 2012; Eshima et al., 2020) have been used in several applications, consisting of allowing practitioners to set prior lexical information, in the form of keywords or labels (Nomoto, 2022), as an attention mechanism to learn robust topics and test hypotheses in a guided fashion, although setting keywords is still considered as a rule-of-thumb exercise (King et al., 2017; Eshima et al., 2020).

While most bag-of-word topic models have been criticized for yielding poor results, due to count data whose structure ignores words' interactions, Steeg and Galstyan (2014) proposed to learn topics from a different perspective using information theory, by computing multivariate mutual information of relevant groups of words that form latent features, known as topics. Correlation Explanation (*CorEx*) (Steeg and Galstyan, 2014) has the advantage of being neither a generative model nor requiring assumptions, but still capable of uncovering meaningful features in diverse applications with sparse data. Extensions of CorEx offer a semi-supervised approach based on predefined keywords, or anchors, that translates the experience or beliefs of practitioners, to learn specific topics as well as hierarchical structures via chained inference (Gallagher et al., 2017).

In network analysis, practitioners usually aim to cluster data into homogeneous groups using several criteria, falling into the class of hierarchical clustering task. Blondel et al. (2008) and Traag et al. (2019) proposed optimized clustering schemes for nonoverlapping features known as *communities*, while Derényi et al. (2005) devised *clique per-*

colation to learn subgraphs as overlapping communities, known as *cliques*, later extended to unweighted and weighted graphs (Farkas et al., 2007). Examples of application in textual analysis considered words as nodes and used communities for document scaling (Diaf, 2023), rhetoric studies (Rule et al., 2015; Bail, 2016) and cliques for an efficient topic detection on short documents (Churchill and Singh, 2020).

If setting keywords improves greatly the quality of the inferred topics (Eshima et al., 2020), it remains mostly an unsupervised task (King et al., 2017), highly dependent on the experience and views of practitioners (Nomoto, 2022), which may show interests in small nested topics not often captured by the models. This work sets an automated anchor strategy, namely *CliqueCorex*, to select keywords, as meaningful features from clique percolation, and to feed them to an anchored CorEx (Gallagher et al., 2017) to maximize topic extraction. Thus, blending two separate clustering schemes into one self-supervised topic model capable of determining keyword groups without human intervention. The number of topics, considered as a hyperparameter, is automatically set to the number of cliques, although users can extend it to learn extra features not coined to any clique.

This strategy frees practitioners from analyzing the corpus in search of relevant keywords and instead automates it by uncovering meaningful mixed-membership cliques, whose semantic structure can be assimilated to powerful subtopics (Ohsawa et al., 1998) emerging from the corpus itself without requiring external prior information. I argue that cliques, as anchors, reinforce the learning process of *CorEx* models by detecting maximally informative latent groups of words, with a preserved semantic structure and an importance order. In other terms, setting semantically-grounded ngrams as prior acts as a semantic regularization to force a more context-based inference.

In many application fields, setting keywords remains a delicate task especially when a word is polysemic or linked to many topics. As for central banking communication, the word *rate* is used in three key measures of monetary policy practices (interest rate, inflation rate and unemployment rate) as well as in other technical terms. Assigning the word *rate* to a unique anchor group may be problematic in probabilistic topic models, as it could later appear in other topics not related the three aforementioned measures, hence lowering the topic quality of the learned models. *CliqueCorex* solves this issue by automatically setting anchors, as mixed-membership nodes, without any constraint on the cliques.

Applied to two different central banking corpora, *CliqueCorex* unfolded granular themes in the Federal Reserve (FED) governors' speeches (1996-2020), where topics revealed the importance of banking supervision and the macroeconomic status in the U.S. central banking discourse, along other secondary, but not less important interests for central bankers as for market competition and innovation. On the European Central Bank speeches (1997-2023), *CliqueCorex* revealed a policy-oriented corpus, closely tied to the ECB missions and objectives, with different interests and reduced topics compared to the FED corpus. In both applications, *CliqueCorex* outperformed standard *CorEx* in terms of total correlation and topic quality.

The paper outlines the build-up of *CliqueCorex* from a network analysis perspective and from statistical learning (Section 2), then implements the proposed algorithm on two central banking corpora (Section 3) and compares them to the standard *CorEx* used by practitioners.

2 Methodology

2.1 Clique Percolation

Uncovering homogeneous groups in a dense, heavily connected network is a difficult task that requires advanced techniques, exceeding the classic clustering methods as for Principal Component Analysis and the K-means.

In network analysis, we define a node as the representation of an entity or a word (Mihalcea and Tarau, 2004), while an edge connects two entities, either directed or indirected. If an edge determines the strength of the link between two nodes, then the network is said to be weighted, otherwise it is unweighted.

We refer to community detection (Fortunato, 2010) the process of identifying strongly connected subgraphs in a given network, usually assimilated to a hard clustering exercise, that assigns each node to just one community (Blondel et al., 2008; Traag et al., 2019), ignoring cases where a node could be shared by many communities, similar to soft clustering. This overlapping feature was devised as *clique percolation* for unweighted (Palla et al., 2005) and weighted (Farkas et al., 2007) graphs.

In text mining, community detection has been used in a nonoverlapping context to better scale documents (Diaf, 2023), extract meaningful word groups (Bail, 2016) or to study lexical shift (Rule et al., 2015), while attempts to use clique percolation targeted topic modeling for small documents (Churchill and Singh, 2020). Nonoverlapping structures are seen as independent features, but several words could belong to different communities, as for the word *united* shared by many entities (United States, United Kingdom, United Nations), hence the necessity to use clique percolation to take into account words' multiple memberships.

As given by Farkas et al. (2007), clique percolation first identifies k-cliques which are fully connected networks with k nodes (starting from k=3) and filters those having theirs intensities higher than a given threshold I. The intensity of a clique C, denoted I_C, is simply the geometric mean of the edge weights w_{ij} associated to the nodes i and j:

$$I_c = \left(\prod_{i<j;i,j \in C} w_{ij} \right)^{(2/k(k-1))}$$

An intermediate step was added to the clique percolation algorithm, known as *CFinder* (Adamcsek et al., 2006), consisting of applying the intensity threshold I to the overlapping cliques, in addition to the already existing k-cliques (Lange, 2021). The challenge here is to optimize both k and I so to not exclude too many nodes and not incorporate many of them. Because percolation assumes the size distribution of communities are following a power-law, the optimal I for each k is just the cutoff above the emergence of a gigantic component (Lange, 2021).

2.2 Correlation Explanation

Most of topic modeling algorithms belong to the generative class (Churchill and Singh, 2020), assuming that documents are generated by a known distribution of terms. Their inference optimizes parameters of topic/term distribution so to maximize likelihood of documents in the dataset over k topics. A popular example is the Latent Dirichlet Allocation (Blei et al., 2003) which sets the basis of most topic models built upon the bag-of-word assumption, with a probabilistic inference that ignores word associations.

Attempts to improve topics' quality led to explore information theory in seeking robust, highly informative groups of words that occur together, without the need to use probabilistic simulations. Steeg and Galstyan (2014) proposed to use *total correlation* (TC) as a measure of mutual information among many variables:

$$TC(X_G) = \sum_{i \in G} H(X_i) - H(X_G)$$

where $H(X) = E_X[-log_2 p(x)]$ is the entropy measure and G denotes a subset of X random variables, in our case words. The total correlation is non-negative and equals zero if and only if the probability distribution factorizes. It could be written as a Kullback-Leibler divergence:

$$TC(X_G) = D_{KL}(p(X_G) || \prod_{i \in G} p(x_i))$$

Searching for latent factor Y, with k possible values, that explains the correlation in X makes the optimization search over all probabilistic functions of X, $p(y|x)$, as :

$$\max_{p(y|x)} TC(X;Y) \text{ s.t. } |Y| = k$$

For m different factors of Y_i, the optimization of *CorEx* (Gallagher et al., 2017) is written as:

$$\max_{G_j, p(y_j|x_{G_j})} \sum_{j=1}^{m} TC(X_{G_j}; Y_j)$$

where Y_j are m binary latent features, or topics, having X_{G_j} as their corresponding groups of word types. Latent factors Y_j are optimized to be informative about dependencies in the data and do not require generative modeling assumptions (Gallagher et al., 2017). Once learned, they can be used iteratively to construct new latent factors in a hierarchical fashion.

The numerical optimization of *CorEx* begins with a randomly initialized parameters, later iteratively updated as for *Expectation-Maximization* algorithm (Gallagher et al., 2017), which adds a binary parameter $\alpha_{i,j}$ equaling one if and only if word X_i appears in topic Y_j (i.e. $i \in G_j$). The previous equation will have its constraint on non-overlapping groups transformed into α:

$$\max_{\alpha_{i,j}, p(y_j|x)} \sum_{j=1}^{m} (\sum_{i=1}^{n} \alpha_{i,j} I(X_i : Y_j) - I(X : Y_j))$$

$$\text{s.t. } \alpha_{i,j} \mathbb{1}[j = arg \max_{j} I(X_i : Y_{\bar{j}})]$$

where $\alpha \in [0, 1]$ is updated at iteration t by $\alpha^t_{i,j} = exp(\lambda^t (I(X_i : Y_j) - \max_{\bar{j}}(X : Y_{\bar{j}})))$

with λ controlling the sharpness of the softmax function.

For *anchored CorEx*, the objective remains the same as for unsupervised *CorEx* but with the inclusion of Z as labels of X, so that the information bottleneck (Gallagher et al., 2017) could be written as:

$$\max_{p(y|x)} \beta I(Z : Y) - I(X : Y)$$

where β controls the trade-off between compressing X and preserving information about the relevance variable Z. Coining a single word X_i to a topic Y_j results in constraining the above optimization scheme by setting $\alpha_{i,j} = \beta_{i,j}$ where $\beta_{i,j} \geq 1$ controls the strength of the anchor. The scheme remains similar if many anchor words are assigned to a given topic (Gallagher et al., 2017).

Algorithm: CliqueCorex

1.Clique detection: Run *clique percolation* algorithm (Farkas et al., 2007) over the network of bigrams and extract k overlapping groups of words, or *cliques*, from bigrams whose occurrence in the corpus is greater than π.

2.Anchored CorEx: The k cliques are used as features to learn an anchored CorEx model (Gallagher et al., 2017) with k topics.

The proposed *CliqueCorex* blends an unsupervised clustering scheme on the corpus' bigrams to uncover cliques as cohesive subgraphs, then use them as robust anchors, or keywords, for the semi-supervised *CorEx*. This frees the application from human intervention and the need of a transfer learning scheme built on external source as for word embeddings (Mikolov et al., 2013; Dieng et al., 2019), specific indexing (Medelyan, 2009) or classifiers (Florescu and Jin, 2018). *CliqueCorex* has the advantage of being naturally tailored to the documents' specification by seeking most informative features in the corpus. Moreover, the number of anchors is expressed as the number of groups emerging from the clique percolation, where each group can have $k \geq 3$ terms, depending on the specifications used (Farkas et al., 2007).

This hybrid scheme transforms the semi-supervised anchored *CorEx* into a fully self-supervised topic model, where the identification of cliques helps relieving the bottleneck when compressing data X into a set of topics Y. Furthermore, the number of topics, equaling the number of retrieved cliques, could be reduced by running another *CorEx* pass, so to build a hierarchical *CorEx* (Gallagher et al., 2017), if the number of cliques is relatively high.

3 Application

3.1 FED speeches

1,488 governor speeches at the U.S. Federal Reserve, during the period 1996-2020, were scraped from the institution's website[1], offering historical developments that accompanied the American monetary authority throughout several episodes and crises over the last three decades. The corpus was lemmatized using *udpipe* model (Straka et al., 2016) to reduce the size of the document-term-matrix and to get robust ngrams when applying clique percolation. This yielded 36 cliques[2], mostly sequences of three words, to be given as keyword groups for the anchored CorEx[3].

Table 2 shows the clique percolation output, consisting of 36 cliques expressing core monetary and macroeconomic interests, along a technical jargon used to describe the economic status and market developments. Cliques contain mostly trigrams and are informative, in a sense that their structure is similar to subtopics. For context-rich terms, as for "*macroeconomic*" and "*inflation*", their associated cliques feature more than 3 terms.

Cliques of Table 2 are used to learn an anchored *CorEx* whose results are shown in Table 4. Topics are ranked by an descending order of importance, based on their contribution to the total correlation (TC), where the first two confirm the importance of banking supervision and macroeconomic stability when communicating about monetary policy in the United States. Efforts of stability and supervision are linked to the post-2008 addresses and interpreted as direct signals toward economic agents (EuropeanParliament et al., 2018).

The remaining topics are a mix of technical (topics 3, 5 and 10) and non-technical topics (topics 6, 7 and 13), the latter consist of a descriptive jargon used in standard economic and financial analyses. Furthermore, the last three topics (34, 35 and 36) having the least contribution to the total correlation are tied to the crisis time 2007-2009, dealing respectively with the housing market, securitization/debt and oil prices.

Noticeable is that *CliqueCorex* outperforms classic *CorEx* in terms of total correlation (Table 1) and topic content (Table 5) whose structure does not prioritize central banking jargon, but rather frequent terms appearing in the corpus. For instance, Topic 1 learned by *CorEx* refers to academic papers and other references used by governors, which similarly

---

[1] `https://www.federalreserve.gov/newsevents/speeches.htm`

[2] The application took into account cliques containing few words, for the sake of illustration.

[3] The learning process for both corpora used anchored *CorEx* with 100,000 iterations and an anchor strength β_{ij}=2

Table 1: Total Correlation (TC) yielded by each model for both corpora.

Model Corpus	Plain CorEx	CliqueCorex
FED speeches	162.63	177.61
ECB speeches	141.01	144.78

appears in the 17th position in the *CliqueCorex* output (Topic 17).

3.2 ECB Speeches

Speeches of executive members at the ECB (2,493 addresses from February 1997 to September 2023) were collected from the ECB website[4] and underwent the same steps to extract cliques as for the FED corpus.

Table 3 shows clique percolation results, that yielded 26 cliques, mostly trigrams and not rich in information as for the FED corpus. Particularly, the ECB corpus does not feature a strong, colorful macroeconomic jargon, but instead stresses out country-related interests because of the monetary union context.

The application of the *CliqueCorex* has a better total correlation than a plain *CorEx* (Table 1) as well as a better topic content, which turned out to be highly policy-oriented and lacking macroeconomic aspects as found in the FED corpus. This is most likely due to the broad interests discussed at the monetary union level, not at a country-specific level as for the FED. For instance, fiscal concerns linked to crises experienced by some country members (Topics 7 and 8 in Table 6) were broadly debated then economic indicators and forecasted aggregates (Topic 12).

Key terms like "*macroeconomic*" and "*inflation*" are not uncovered in the clique percolation step and only "*inflation*" is captured later by *CliqueCorex* in a more structural context involving unemployment and growth (Topic 21 in Table 6), but far less important then other descriptive topics.

Moreover, the corpus contains small multilingual paragraphs used by some speakers during their addresses (Topic 17 in Table 6 and Topic 2 in Table 7) and specific interests other than its main mission of price stability.

4 Conclusion

Probabilistic topic models continue to be the go-to solution when dealing with textual data under its different aspects. Behind their popularity, they are still limited by the fact that they consider words as independent features, in addition of having a probabilistic learning process based on word counts, yielding poorly informative results. While numerous extensions were developed to improve topic extraction, developments based on information theory suggest model-free algorithms capable of yielding superior topic quality, as for *correlation explanation*, and offering semi-supervised extensions for keywords and labels. This paper proposed the use of *clique percolation*, as a pre-processing step, enabling the automatic identification of anchors as *cliques* for a fully-automated anchored *CorEx*, encompassing neither external information nor human intervention. The resulting blend, named *CliqueCorex*, is a self-supervised topic model built on subgraphs, assumed to be subtopics from the corpus' bigram network, yielding maximally informative binary topics when blended to an anchored *CorEx*.

By adopting cliques as a semantic regularization scheme, *CliqueCorex* proved a higher ability in capturing hidden topics and other features overlooked by non-guided topic models. On two central banking corpora, known to have a rich imbricated context, *CliqueCorex* demonstrated a robustness in unfolding deep interests in monetary policy practices and reveal their relative importance, with rich monetary policy-oriented topics found at the U.S. Federal Reserve addresses, while policy-oriented interests dominate the speeches given at the European Central Bank, but not necessarily monetary or macroeconomic ones.

Clique percolation first revealed different scopes of interests the corpora have, although both deal with monetary policy, and the necessity to set tailored anchors for each corpus. This reinforces the claim that transferring keywords or anchors within the same task is not always indicated to extract nested features.

Uncovered cliques, in addition of acting as subtopics because of their semantic structures, fitted perfectly the anchored *CorEx* mechanism to deliver cohesive topic contents and reveal interesting corpus' orientation, in terms of topic content and importance.

---

[4] `https://www.ecb.europa.eu/press/key/html/downloads.en.html`

Table 2: Cliques in the FED corpus

Clique	Words
1	bank thrift supervision
2	macroeconomic policy stability objective
3	economic growth prospect
4	free trade flow
5	long time horizon
6	less well able
7	let now turn
8	dual mandate objective
9	climate risk relate
10	term treasury yield
11	across many market
12	achieve domestic inflation run trend goal objective
13	one part time
14	one recent study year
15	debt service burden
16	across national border
17	discussion paper series
18	put upward pressure
19	early first half last next past several ten three time twenty two week year
20	core pce price
21	committee fomc member participant
22	another important key reason way area
23	real short term time
24	government guarantee program
25	federal government spending
26	consumer financial protection
27	reduce regulatory burden
28	find new way
29	asset liability side
30	even far great
31	base capital measure
32	american community economic
33	develop new world
34	consumer durable good
35	agency debt issue security mbs
36	high oil price

Table 3: Cliques in the ECB corpus

Clique	Words
1	full information set
2	just mention two
3	precise quantitative definition
4	policy relevant horizon
5	billion euro banknote coin
6	content presentation slide
7	automatic fiscal stabiliser
8	greece ireland portugal
9	france germany italy
10	credit loss provision
11	conference discussion paper series
12	commission economic forecast
13	area non resident
14	available empirical evidence
15	already know well
16	common different objective set
17	council decide last meet
18	analysis can find
19	asset portfolio allocation
20	annual data report requirement
21	high structural unemployment
22	council decision make take
23	general term orientation
24	single technical platform
25	become self evident
26	become fully operational

Table 4: *CliqueCorex* on the Federal Reserve speeches

Topic	Top Words
1	supervision, regulator, supervisor, supervisory, banking, regulation, institution, deposit, oversight, bank
2	stability, macroeconomic, policy, shock, central, international, policymaker, objective, crisis, implication
3	productivity, growth, output, labor, production, gdp, prospect, worker, boost, wage
4	trade, flow, free, exchange, foreign, dollar, denominate, asia, currency, asian
5	horizon, argue, long, theory, empirical, time, weight, variable, argument, optimal
6	less, able, well, same, without, amount, net, account, only, both
7	turn, now, let, then, think, second, give, out, after, begin
8	mandate, dual, objective, laubach, bind, woodford, deviation, curve, reifschneider, jackson
9	risk, relate, certain, must, subject, procedure, whether, function, allow, exercise
10	treasury, yield, term, fund, return, normal, bond, condition, maturity, back
11	across, many, result, example, market, process, exist, require, significant, limit
12	inflation, run, outlook, monetary, employment, unemployment, nominal, trend, consumption, forecast
13	time, part, one, if, under, could, some, activity, case, because
14	recent, factor, percent, study, datum, one, compare, indicate, year, survey
15	service, industry, technology, customer, competitive, competition, innovation, technological, marketplace, efficiency
16	border, across, national, cross, country, globalization, among, infrastructure, nation, western
17	paper, pp, series, vol, journal, economics, pdf, discussion, cambridge, washington
18	pressure, upward, put, downward, demand, ease, japan, recovery, cut, stimulus
19	early, three, two, past, first, half, several, time, last, year
20	price, pce, core, estimate, indicator, historical, food, gradual, projection, solid
21	fomc, committee, open, accommodation, target, maximum, stance, incoming, path, easing
22	key, approach, area, discuss, reason, framework, analysis, practice, specific, implement
23	short, term, real, suggest, appear, relative, period, likely, somewhat, time
24	government, guarantee, program, reform, taxpayer, brothers, bankruptcy, street, suffer, serious
25	spending, expenditure, fall, household, percentage, down, index, quarter, recession, sustainable
26	loan, lending, borrower, credit, protection, lender, access, mortgage, consumer, commercial
27	regulatory, requirement, propose, organization, disclosure, proposal, compliance, examination, profile, rulemaking
28	find, way, question, problem, new, form, try, often, go, know
29	asset, liability, portfolio, investor, sheet, liquid, fail, instrument, against, arise
30	far, even, seem, indeed, great, moreover, little, still, quite, hence
31	measure, capital, thus, however, potential, base, reflect, generally, substantial, expect
32	community, american, education, family, school, training, americans, million, educational, census
33	world, develop, century, new, modern, history, society, power, dramatic, yet
34	consumer, home, homeowner, housing, income, residential, construction, foreclosure, homeownership, affordable
35	liquidity, agency, debt, loss, security, funding, counterpartie, stress, default, securitization
36	price, oil, decline, rise, sharp, high, supply, above, low, tighten

Table 5: Plain *CorEx* on the Federal Reserve speeches

Topic	Top Words
1	pp, vol, economics, journal, cambridge, david, massachusetts, nber, papers, university
2	spending, output, decline, rise, slow, boost, gdp, consumption, fall, labor
3	supervisor, supervisory, regulator, supervision, regulatory, regulation, oversight, exposure, banking, institution
4	fomc, inflation, outlook, monetary, employment, unemployment, expectation, nominal, stance, maximum
5	liquidity, systemically, funding, stress, severe, crisis, macroprudential, collateral, vulnerability, repurchase
6	text, pdf, speech, washington, governors, april, www, november, march, february
7	outsourcing, amy, needs, send, thankfully, retraining, multinational, involuntarily, saez, ashenfelter
8	organization, establish, transaction, develop, create, legal, act, protection, address, effort
9	expect, balance, risk, measure, potential, firm, asset, activity, sheet, generally
10	reduce, effect, relatively, short, investor, far, lead, capital, substantial, increase
11	practice, certain, apply, problem, procedure, limit, rule, issue, subject, set
12	macroeconomic, theory, policymaker, empirical, macroeconomics, taylor, equilibrium, influence, economist, variable
13	international, united, european, states, japan, foreign, exchange, central, global, currency
14	raghuram, rajan, gaps, makeup, text5, text6, text8, text7, text9, firms
15	education, educational, school, young, college, life, skill, americans, adult, population
16	production, productivity, equipment, worker, half, living, producer, inventory, war, fast
17	loan, borrower, lending, lender, mortgage, credit, commercial, underwriting, subprime, securitization
18	keys, toggle, caption, mediainfo, xs, transcripttext, transcriptlinkurl, col, fullscreen, myplayer
19	approach, framework, implement, assessment, appropriate, guidance, quantitative, conduct, consider, model
20	question, think, answer, reason, fact, argument, political, try, hand, precisely
21	housing, income, family, home, household, homeowner, residential, construction, homeownership, survey
22	shock, uncertainty, run, imply, term, normal, episode, movement, uncertain, response
23	suggest, period, argue, early, evidence, view, course, factor, adjust, quite
24	century, old, free, society, generation, nineteenth, history, virtually, twentieth, revolution
25	process, individual, information, involve, enhance, effective, ability, use, recognize, responsibility
26	technology, computer, technological, electronic, internet, service, network, automate, telecommunication, physical
27	actor, head, boivin, bridgewater, scrapping, scandinavian, risks, uncorrelated, confidently, vestin
28	debt, investment, finance, bond, borrowing, private, borrow, dollar, collapse, saving
29	audit, auditor, privacy, laundering, payments, sarbanes, oxley, sponsoring, treadway, merchant
30	analysis, example, datum, include, base, determine, particularly, study, relationship, relate
31	census, urban, local, racial, metropolitan, finances, barrier, resident, hispanic, hispanics
32	clive, bring, timmermann, penalver, billi, exploration, farmer, matheson, sandri, cardia
33	competitive, industry, competition, innovation, law, marketplace, efficient, opportunity, efficiency, competitor
34	hear, fed, listen, chair, district, communications, hope, proud, president, prepare
35	error, rational, rigidity, rose, agent, al, manuscript, et, override, linear
36	increasingly, facilitate, pricing, expand, availability, cash, pay, profitable, segment, hedge

Table 6: *CliqueCorex* on the ECB speeches

Topic	Top Words
1	information, set, introduction, regard, order, respect, importance, practice, specific, conduct
2	mention, two, seem, try, difficult, maker, experience, agent, another, world
3	definition, quantitative, precise, variable, reference, reserve, strategy, signal, influence, interpret
4	horizon, response, policy, underlie, relevant, volatility, imply, recent, associate, broad
5	banknote, coin, cash, changeover, circulation, card, store, january, dollar, electronic
6	slide, content, presentation, annex, kb, pdf, peter, flatten, download, proxy
7	fiscal, deficit, stabiliser, reform, pact, automatic, budget, sustainable, government, competitiveness
8	ireland, portugal, greece, spain, education, mobility, young, cyprus, belgium, slovenia
9	germany, italy, france, german, di, age, inequality, discourage, italian, five
10	credit, loss, systemic, crisis, banks, supervisory, supervision, liquidity, exposure, provision
11	paper, journal, vol, pp, economics, series, nber, university, research, al
12	forecast, economic, average, period, factor, gdp, estimate, percentage, indicator, hicp
13	non, investment, large, capital, reduce, small, united, turn, less, total
14	empirical, evidence, suggest, relative, effect, aggregate, shock, likely, premium, cycle
15	know, think, often, even, little, say, go, already, question, well
16	different, set, objective, common, principle, problem, framework, rule, define, task
17	decide, council, die, le, ich, der, la, zu, und, meet
18	analysis, find, theory, federal, model, argue, academic, behaviour, economist, bubble
19	asset, lend, portfolio, loan, sheet, bond, fund, maturity, household, yield
20	data, report, type, source, requirement, publish, distribution, disclosure, annual, sample
21	unemployment, inflation, structural, outlook, wage, decline, pressure, growth, low, rate
22	treaty, decision, independence, maastricht, council, union, state, mak, responsibility, institutional
23	general, term, short, degree, however, development, rather, similar, extent, although
24	border, single, platform, technical, infrastructure, field, service, efficient, payment, integration
25	become, self, cause, back, face, put, consequence, prevent, happen, bad
26	become, fact, example, role, base, reason, limit, fully, consider, conclusion

Table 7: Plain *CorEx* on the ECB speeches

Topic	**Top Words**
1	journal, paper, pp, economics, vol, nber, university, american, literature, al
2	die, der, und, ich, le, zu, je, la, auf, une
3	even, different, fact, seem, less, rather, little, case, reason, often
4	inflation, outlook, pressure, medium, wage, expectation, decline, projection, upward, uncertainty
5	relative, suggest, evidence, effect, factor, empirical, aggregate, likely, short, tend
6	rate, growth, structural, low, grow, gdp, inter, fiscal, condition, price
7	political, always, go, simply, idea, world, true, bad, must, perhaps
8	recovery, purchase, accommodative, household, ease, return, negative, stimulus, accommodation, fall
9	sheet, lend, fund, banks, loan, liquidity, crisis, credit, maturity, sovereign
10	leverage, asset, loss, response, episode, risk, macro, mitigate, boom, buffer
11	infrastructure, payment, transaction, user, settlement, provider, service, retail, field, initiative
12	pandemic, covid, lagarde, coronavirus, christine, digital, op, pepp, carbon, green
13	academic, theoretical, understand, press, economist, hypothesis, assumption, attempt, keynesian, conference
14	supervision, supervisory, systemic, regulatory, supervisor, basel, prudential, regulation, exposure, management
15	introduction, development, regard, general, thus, final, several, various, particular, mention
16	series, estimate, percentage, chart, income, bulletin, zero, survey, occasional, total
17	labour, productivity, export, labou, population, worker, capita, deficit, competitiveness, education
18	currency, banknote, changeover, exchange, coin, quot, accession, shall, enlargement, erm
19	value, refer, available, amount, type, use, participant, feature, form, wide
20	framework, authority, institution, principle, rule, task, procedure, implementation, set, establish
21	definition, analysis, reference, information, variable, conduct, quantitative, assessment, strategy, appropriate
22	integration, border, competition, cross, single, integrate, efficient, barrier, europe, transfer
23	treaty, pact, independence, emu, maastricht, credibility, institutional, union, decision, responsibility
24	fan, gov, villier, eride, ottaviano, wogau, hicpx, smile, disorder, spot
25	technology, happen, get, big, revolution, pay, online, protect, stop, era
26	gonzález, páramo, manuel, josé, pedersen, reluctant, responses, sifi, workshop, metrick

References

Balázs Adamcsek, Gergely Palla, Illés J. Farkas, Imre Derényi, and Tamás Vicsek. 2006. CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023.

Edoardo M. Airoldi, David Blei, Elena A. Erosheva, and Stephen E. Fienberg, editors. 2014. *Handbook of Mixed Membership Models and Their Applications*. Chapman & Hall / CRC Handbooks of Modern Statistical Methods. Taylor and Francis, Hoboken.

Christopher A. Bail. 2016. Combining natural language processing and network analysis to examine how advocacy organizations stimulate conversation on social media. *Proceedings of the National Academy of Sciences*, 113(42):11823–11828.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120, New York, NY, USA. Association for Computing Machinery.

David M. Blei and Jon D. McAuliffe. 2007. Supervised topic models. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 121–128, Red Hook, NY, USA. Curran Associates Inc.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Rob Churchill and Lisa Singh. 2020. Percolation-based topic modeling for tweets. *WISDOM 2020 : The 9th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining*.

Rob Churchill and Lisa Singh. 2022. The evolution of topic modeling. *ACM Comput. Surv.*, 54(10s).

Imre Derényi, Gergely Palla, and Tamás Vicsek. 2005. Clique percolation in random networks. *Phys. Rev. Lett.*, 94:160202.

Sami Diaf. 2023. CommunityFish: A Poisson-based document scaling with hierarchical clustering. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 59–67, Online. Association for Computational Linguistics.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Shusei Eshima, Kosuke Imai, and Tomoya Sasaki. 2020. Keyword assisted topic models. *arXiv preprint arXiv:2004.05964*.

EuropeanParliament, Directorate-General for Internal Policies of the Union, P Hubert, and C Blot. 2018. *Central bank communication during normal and crisis times – Monetary dialogue September 2018 – In-depth analysis*. European Parliament.

Illés Farkas, Dániel Ábel, Gergely Palla, and Tamás Vicsek. 2007. Weighted network modules. *New Journal of Physics*, 9(6):180.

Corina Florescu and Wei Jin. 2018. Learning feature representations for keyphrase extraction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Santo Fortunato. 2010. Community detection in graphs. *Physics Reports*, 486(3):75–174.

Ryan J. Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Alan Greenspan. 2004. Risk and uncertainty in monetary policy. *American Economic Review*, 94(2):33–40.

Thomas Griffiths, Michael Jordan, Joshua Tenenbaum, and David Blei. 2003. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213, Avignon, France. Association for Computational Linguistics.

Gary King, Patrick Lam, and Margaret Roberts. 2017. Computer-assisted keyword and document set discovery from unstructured text. *American Journal of Political Science*, 61(4):971–988.

Jens Lange. 2021. Cliquepercolation: An r package for conducting and visualizing results of the clique percolation network community detection algorithm. *Journal of Open Source Software*, 6:3210.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K. Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 81–88.

Olena Medelyan. 2009. *Human-competitive automatic topic indexing*. Ph.D. thesis, Citeseer.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tadashi Nomoto. 2022. Keyword extraction: a modern perspective. *SN Computer Science*, 4(1):92.

Yukio Ohsawa, Nels E. Benson, and Masahiko Yachida. 1998. Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor. *Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries -ADL'98-*, pages 12–18.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.

Alix Rule, Jean-Philippe Cointet, and Peter S. Bearman. 2015. Lexical shifts, substantive changes, and continuity in state of the union discourse, 1790–2014. *Proceedings of the National Academy of Sciences*, 112(35):10837–10844.

Greg Ver Steeg and Aram Galstyan. 2014. Discovering structure in high-dimensional data through correlation explanation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 577–585, Cambridge, MA, USA. MIT Press.

Milan Straka, Jan Hajič, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1).

Double Decoder: Improving latency for Streaming End-to-end ASR Models

Riqiang Wang[*], **Shreekantha Nadig**[*], **Daniil Kulko, Simon Vandieken,**
Chia-Tien Chang, Seyyed Saeed Sarfjoo, Jonas Robertson
Dialpad, Canada
`{riqiang.wang, shreekantha.nadig, daniel.kulko, svandieken,`
`karol.chang, saeed.sarfjoo, jonas}@dialpad.com`

Abstract

In this paper, we propose a novel decoding algorithm for streaming End-to-end (E2E) automatic speech recognition (ASR) models, the double decoder. By comparing it with existing decoding algorithms, we argue that this new method achieves a better balance between word error rate, latency and streaming stability, notably by reducing latency without WER degradation but with degradation in stability. The algorithm also does not require any change in model weights. We show results on a Conformer-CTC model trained on the LibriSpeech dataset, which indicates that the proposed double decoder maintains the same WER as buffered decoding while reducing the latency by the size of the look-ahead used in decoding. We also show that the proposed method is generalizable. For example, we apply it to the Zipformer-CTC-Transducer model, which traditionally uses the default decoding, and it achieves better WER and latency at the expense of increased computational cost.

1 Introduction

Real-time automatic speech recognition (ASR) systems are a critical part of many of the industrial speech understanding applications. For example, delays in the ASR cause delays in downstream natural language processing (NLP) tasks. Other factors like accuracy, measured by word error rate (WER), and streaming stability (Shangguan et al., 2020a) are also important factors to ensure a good overall user experience in real-time language processing systems.

While deep neural network-hidden Markov model (DNN-HMM) hybrid ASR models have traditionally been used for efficient streaming ASR inference, many advancements in end-to-end (E2E) ASR research have shown that E2E models have more performance potential. For instance, the connectionist-temporal-classification (CTC) loss (Graves et al., 2006) models longer context without pre-defined alignment and allows for low frame-rate decoding (Pundak and Sainath, 2016). RNN-Transducers (RNN-T) (Graves, 2012) jointly train an internal language model with an acoustic model, further improving the modelling capabilities of a single model. Attention encoder-decoder (AED) models such as the Listen-Attend-Spell (LAS) (Chan et al., 2016) models exploit the attention mechanism to model much longer context. With the introduction of Transformer (Vaswani et al., 2017) and its variants (Gulati et al., 2020), E2E models have been pushing the state-of-the-art WER on various ASR datasets (Chen et al., 2023).

However, for streaming ASR, the search for a good trade-off between latency, stability and accuracy remains an open problem. The aforementioned E2E models achieve good WER by incorporating more audio context, and exploiting more parameters, translating to higher latency and more compute cost. For example, an AED model for ASR requires the entire input sequence before starting to generate output tokens; the Transformer model uses absolute positional embeddings, limiting its applicability to streaming real-time ASR systems (Dai et al., 2019).

In our study, we propose an algorithm we call the *double decoder*. In essence, we run the decoder twice on the encoder outputs - once on the *look-ahead* or the most recent chunk of audio to speculatively display low latency results, then once on the chunk behind it, with a delay. No change in the weights of the model is needed. The algorithm builds on the existing buffered decoding method, which is designed for inference time and addresses the limitation of Transformers, which typically require the complete sequence for inference.

This simple algorithm has not been published at the time of writing and we believe many ASR engineers can benefit from this method. We show that our proposed method can improve the suitability of

---

[*] Equal contribution.

E2E models for streaming ASR by achieving a better balance between latency, accuracy, and stability. In particular, it enables streaming for Conformer-CTC (Graves et al., 2006; Gulati et al., 2020) and other CTC models, which would otherwise have undesirable performance for streaming. This is achieved by reducing the latency while keeping the WER of the buffered decoding method. We also show an example with the Zipformer-CTC-Transducer model where we reduce the WER and latency when we use the right most part of the original chunk size as look-ahead.

2 Traditional Decoding Algorithms for Streaming

There are a few assumptions and definitions to clarify for the description of decoding algorithms. For simplicity, we assume there is no sub-sampling so the model stride at output time is the same as the input. We also define *model* to be the neural network outputting log probabilities for each frame, whereas the *external decoder* is the external decoding algorithm, not to be confused with the decoder inside an AED model. Its forward function takes the log probabilities from the model given the latest input chunk, and outputs the text hypothesis for the entire audio so far. It updates its internal state for caching the history: for greedy decoding, the latest text output is appended to the text history; for beam search, the beam gets updated.

2.1 Default Method

The most straightforward default method is to decode using the same duration for input length and step size. This can be sufficient if the model incorporates an RNN component, where the left context is inherently represented in the RNN cell state. In this method, no computation is wasted as every input frame passes through the model exactly once. Conceptually, Figure 1a shows this decoding algorithm. At each time step t, the model reads input chunk x_t and outputs the log probabilities, and the external decoder processes them into text.

2.2 Buffered Decoding

For Transformer or Transformer-like models such as the Conformer (Gulati et al., 2020), every layer attends to the exact same context as the input audio. It does not incorporate history context directly, therefore we need to explicitly include history audio h_t besides x_t as input. Moreover, we have seen in the literature (Moritz et al., 2020) as well as empirically that adding the right context (look-ahead l_t) improves WER. This is called buffered decoding as we need to keep h_t, x_t and l_t together in a buffer, run the model on everything at each time frame and only keep the log probabilities for x_t. This method has been the default streaming method for Transformer-like models[1], and it is illustrated in Figure 1b. In practice, l_t is the actual latest chunk. To only output the results for x_t means a constant additional delay of $|l_t|$ after getting x_t.

3 Proposed Decoding Algorithm

Our proposed method, the *double decoder*, builds on the buffered decoding with one simple modification: we use a temporary decoder on top of the main external decoder. For each time step t, the model produces output log probabilities p_t for h_t, x_t and l_t, where only the log probabilities given x_t are passed into the main external decoder. The state of the external decoder gets updated, then it is copied to be the state of the temporary decoder. Subsequently, the log probabilities given l_t are decoded by a temporary decoder to produce the partial hypothesis y_t, which is the text hypothesis given in the audio so far. The partials get replaced with new ones at each time step. See Algorithm 1 and Figure 1c.

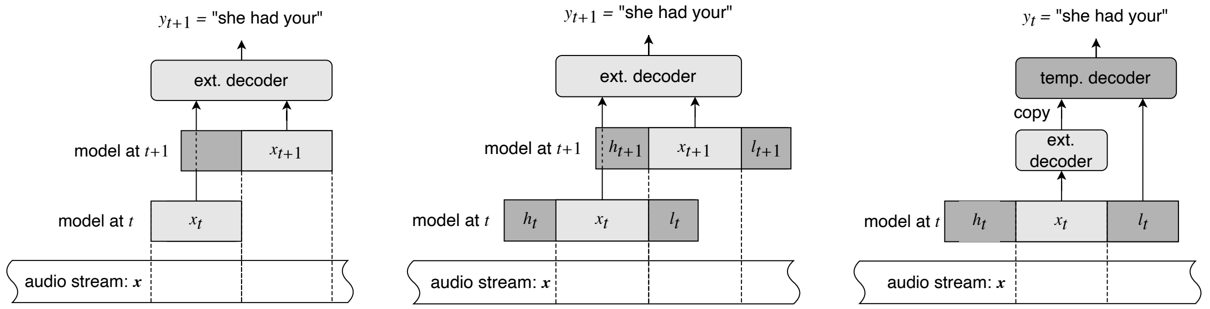
Algorithm 1: Double Decoder

```
input  : audio stream $\boldsymbol{x}$, same $|h_t|$, $|x_t|$,
         $|l_t|$ for every $t$
output : partial hypotheses $\boldsymbol{y}$
Initialize model, ext_decoder, $t \leftarrow 0$;
while $\boldsymbol{x}$ not ended:
    Get latest chunks $h_t$, $x_t$, $l_t$ from $\boldsymbol{x}$;
    $p_t \leftarrow$ model.forward($h_t + x_t + l_t$);
    $t'_0 \leftarrow$ start time of $x_t$;
    $t'_1 \leftarrow$ end time of $x_t$;
    ext_decoder.forward($p_t$ from $t'_0$ to $t'_1$);
    temp_decoder $\leftarrow$ copy(ext_decoder);
    $y_t \leftarrow$ temp_decoder.forward($p_t$ from $t'_1$ onward);
    $t \leftarrow t + |x_t|$;
```

Using this algorithm, we ensure that the final

---

[1] Example code using the NeMo toolkit at https://github.com/NVIDIA/NeMo/tree/main/examples/asr/asr_chunked_inference



(a) Default decoding: decoding each current chunk

(b) Buffered decoding: decoding with look-ahead

(c) *Double decoder*: using a temporary decoder for outputting the look-ahead

Figure 1: Three decoding methods. The chunks x_t and the context h_t and l_t represents what the mdoel (encoder) has access to, and we skip the steps where the model encodes the audio and output log probabilities. The *copy* action copies the internal state of the external (ext.) decoder to the temporary (temp.) decoder, which includes the information about the history. Notice that y_t from the *double decoder* outputs the same text for y_{t+1} from the other two methods. This is to demonstrate that the *double decoder* can output the same text earlier than the other methods. Light grey colour means the result is cached and used for the final output, while dark grey means the content is discarded after one time step.

hypothesis is made up of the outputs given x_t where the accuracy is better. The hypothesis given the look-ahead l_t is only displayed but never saved into the main decoder's state. At the same time, we avoid waiting for getting l_t purely as context, compared to buffered decoding.

4 Related Work

Many previous studies on streaming ASR have focused on improving the internal efficiency of the models. Modifications to the Transformer are proposed to make it more suited for real-time streaming ASR - this includes relative positional embeddings (Shaw et al., 2018), caching and reusing intermediate network states (Dai et al., 2019), limiting the look-ahead acoustic context (Noroozi et al., 2024) and using time-restricted self-attention (Moritz et al., 2020). Most of the proposed methods require changes to the original ASR architecture and re-training of the model, whereas our proposed method works on pre-trained models.

Other studies take a more holistic view of optimizing for streaming ASR, considering the trade-off between context size, latency, accuracy, and streaming stability, i.e. whether the displayed words gets revised as the transcript becomes more complete (Bruguier et al., 2016; Moritz et al., 2020; Shangguan et al., 2020a,b, 2021). In Shangguan et al. (2020a), the authors proposed increasing partial emission latency, unifying text normalization for different domains and using domain ID to improve the stability of partials. Whereas in Bruguier et al. (2016), the authors proposed an algorithm to select more stable hypotheses during beam search decoding. We take inspiration from these studies and analyze the stability of our proposed method.

In terms of architectures used for streaming, recent studies of streaming ASR have focused on improving RNN-T models as they showed superior WER performance. To further improve the final WER, earlier efforts use a second pass LAS model to do beam search or to rescore the RNN-T hypotheses (Sainath et al., 2020, 2019). He et al. (2019); Narayanan et al. (2021); Sainath et al. (2021); Shangguan et al. (2020b) use cascaded encoders, with a causal encoder which passes its output to another non-causal encoder. This eliminates the need for training a separate re-scoring model, unifying streaming and non-streaming models while improving the final transcription. Related, Yu et al. (2021) presented a more detailed exploration of a unified streaming and non-streaming RNN-T model.

Our proposed method is similar to the cascaded encoders (Narayanan et al., 2021), besides the fact that we make use of only a single model with a single encoder, saving the compute of running an extra encoder. It can be seen as a special case of cascaded encoders where the second encoder is just copying the embeddings from the first one. However, since we are using only one model, the final WER is bound by the offline model performance, whereas the cascaded encoders' second pass improves WER.

A similar idea of fast-slow two-head decoding is explored for older hybrid ASR in Li et al. (2020). However, Li et al. (2020)'s method also involves changing the model architecture and training a second encoder. Our method is purely a decoding algorithm. It can be argued that their method is suitable for LSTM models without explicit access to look-ahead data, and our method is preferable for Transformer-like E2E models.

Finally, as many of the techniques mentioned above focus on RNN-T, we show that our decoding algorithm can be applied to not only RNN-Ts but also CTC models. Models trained with CTC are typically outperformed by RNN-T or AEDs in WER, but they have a lower real-time factor (RTF) (Zhang et al., 2021), making them suitable candidates for streaming ASR where latency is crucial.

5 Evaluation Metrics

5.1 Accuracy

We use WER to measure the accuracy performance of the streaming inference. For WER, if we use the same context (buffer) size, we expect the WER to be the same for buffered decoding and *double decoder*, since the chunks used to generate the final hypothesis are exactly the same.

5.2 Streaming Stability

To discuss streaming stability, we first define the relevant terms used: A partial hypothesis is a text sequence outputted by the ASR system before the end of an utterance is reached. A final hypothesis is produced after end-pointing - either the end of the audio is reached or the end-pointer detects the boundary of the utterance. As we replace previous partials on the display with new partials, the words may get revised which can be seen as the effect of instability. It should be noted that we do not measure the WER of partial hypotheses for the stability since both the reference and the hypothesis are incomplete and changing.

We employ the unstable partial word ratio (UPWR) introduced in Shangguan et al. (2020a) to measure instability. To calculate UPWR, we sum up the revised or unstable tokens in each partial hypothesis when compared to the next partial hypothesis, then divide it by the number of tokens in the final. The closer UPWR is to 0, the more stable the system. As an example, for the partial hypotheses produced by the *double decoder* in Table 1, we have three unstable tokens, *but*, *please* and *him*. The final hypothesis has 10 tokens, therefore UPWR = 0.3. It should be noted that we cannot directly compare the results of partial stability with the original paper, since they use formatted transcripts.

As we compare with the buffered decoding method, we expect stability to degrade, since we are incorporating more speculative text outputs in the hypothesis. As the buffered decoding method always takes the middle chunk which forms the final hypothesis, we expect the UPWR to be almost 0. On the other hand, *double decoder* will produce mostly non-zero scores.

5.3 Latency

To simplify the calculation of latency differences, we consider three parts of latency that make up the total latency in a continuous, streaming ASR system, presuming I/O and other system latency to be negligible. Firstly, there is the delay for accumulating audio stream T_a, composed of $|l_t + x_t|$ which makes up a constant delay. It should be noted that the history size $|h_t|$ corresponds to a one-time delay at the start of the audio stream. However, by padding the beginning of the audio with artificial silence, we can effectively reduce this delay to zero.

Furthermore, for every audio input x, we consider the model forward latency $T_m(x)$, which is the inference time for producing frame-wise log probabilities. Independently, we consider decoding latency $T_d(x)$, which is the time taken by the external decoder to produce one partial hypothesis, given the probabilities.

In this context, we can calculate the theoretical latency of buffered decoding as $|l_t + x_t| + T_m(h_t + x_t + l_t) + T_d(x_t)$. Given our algorithm, if we compare it with the buffered decoding approach, we can see that by outputting the hypotheses for the right contexts of each time step, we reduce the theoretical latency to $|x_t| + T_m(h_t + x_t + l_t) + T_d(x_t + l_t)$, where $\Delta T = |l_t| - T_d(l_t)$. We show ΔT in Section 7.

6 Experiment Setup

We have chosen two models to illustrate the effectiveness of the algorithm with or without explicit historical context. First we apply it to the small Conformer-CTC model from NVIDIA NGC[2] to

---

[2] `https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_ctc_small_ls`

Table 1: Sample comparison of partials generated by buffered decoding vs. double decoder decoding. It is taken from Test-clean and the context size is 1.2 seconds.

Buffered partial hypotheses	*Double decoder* partial hypotheses
	i never
i never knew	i never knew of
i never knew but	i never knew but
i never knew but one ma	i never knew but one man
i never knew but one man who coul	i never knew but one man who could ever
i never knew but one man who could ever pleas	i never knew but one man who could ever please him
i never knew but one man who could ever pleasing	i never knew but one man who could ever pleasing

show its effectiveness comparing to buffered decoding. The model has 13M parameters. For the decoding method, we utilize a streaming beam search decoder built on PyCTCdecode[3]. For the beam search, we use a 3-gram language model (LM) with LM weight of 0.2, insertion penalty of 0.3. The beam width is set to 100, and max active tokens is 20. When we decode with buffered decoding or the *double decoder*, the middle chunk $|x_t|$ is 0.6 s. We evaluate WER, latency and streaming stability for Conformer-CTC given different context sizes.

We also show that our algorithm can be applied to a Zipformer-CTC-Transducer model to improve its WER against default decoding. We trained a 66M parameter Zipformer-medium(Yao et al., 2024) streaming cache-aware model. The final loss for the model was computed as a weighted sum of a CTC (Yao et al., 2023) and pruned RNN-T loss (Kuang et al., 2022) using K2[4] with CTC weight of 0.2. The decoding method for this model is greedy decoding. The training recipe can be found on the Icefall repository[5]. The model does not require history input, i.e. $|h_t| = 0$. Instead, it keeps a cached state of intermediate layers for the past history.

Both models are trained on the 960 hours of LibriSpeech data(Panayotov et al., 2015), and are evaluated on LibriSpeech test-sets. Latency is computed on a virtual machine using intel Cascade Lake CPUs.

7 Results

7.1 Results on Conformer-CTC

Table 2 shows the WER for both buffered decoding and *double decoder*, as well as the latency difference between the two decoding methods. As discussed in Section 5.1, WER is the same for buffered decoding and *double decoder*. We see that on LibriSpeech Test-clean and on Test-other, WER decreases by approximately the same rate. This demonstrates the benefit of using longer context, whose effect is consistent across different testing conditions.

Regarding latency, as discussed in Section 5.3, *double decoder* reduces latency from buffered decoding by $|l_t| - T_d(l_t)$. We can clearly see from Table 2 $T_d(l_t)$ is an order of magnitude smaller than the look-ahead size $|l_t|$, and does not increase linearly with $|l_t|$. Therefore, we can conclude that with the studied context size the *double decoder* always provides a latency reduction of approximately the same duration as the look-ahead size. As the WER decreases when the context size increases, we can effectively achieve better WER while maintaining latency, or reduce latency while maintaining WER, by using the *double decoder*.

Figure 2 and 3 shows the UPWR results on Test-clean and Test-other. We confirm that using the *double decoder* results in higher UPWR scores. However, there are other interesting trends with regards to partial stability. Firstly, we observe a downward trend for the buffered decoding as we increase the context size. This indicates an improvement in the quality of the middle chunk. At the same time, UPWR for *double decoder* decreases in Test-clean, but increases in Test-other, as we increase the context size. This indicates that in noisy conditions, the lookahead becomes much more unstable, reflected both in raw UPWR score and its variation. As the acoustic condition in real-life ASR applications are not always clean, this trend suggests that we cannot blindly increase the context size, but we need to combine this metric together with latency and WER for hyper-parameter tuning.

Another interesting difference between the two

---

[3] `https://github.com/kensho-technologies/pyctcdecode`

[4] `https://github.com/k2-fsa/k2`

[5] `https://github.com/k2-fsa/icefall/tree/master/egs/Librispeech/ASR/zipformer`

Table 2: WER (both buffered and *double decoder* have the same value) for the Conformer-CTC given different context sizes. $|x_t| = 0.6$ s. $|h_t|$ and $|l_t|$ are shown in the table. Latency reduction by employing *double decoder* compared to buffered decoding is calculated by $|l_t| - T_d(l_t)$ as shown in Section 5.3.

Context size (s)	$\|h_t\|$, $\|l_t\|$ (s)	Test-clean (%)	Test-other (%)	$T_d(l_t)$ (ms) / 95% CI
1.2	0.28, 0.32	11.84	23.76	4.4 / [3.1,5.2]
1.8	0.56, 0.64	5.61	13.65	3.4 / [1.9,4.0]
2.4	0.88, 0.92	4.53	11.22	**3.0 / [1.8,3.6]**
3.0	1.2, 1.2	4.04	10.06	3.0 / [1.8,3.7]
4.0	1.68, 1.72	**3.6**	**9.01**	3.2 / [1.8,4.3]

decoding algorithms becomes clear as we look into the raw partial hypotheses in text. In Table 1, we can clearly see that buffered decoding produces unfinished non-word tokens such as *ma*, *coul*. On the other hand, the unstable tokens produced by the *double decoder* are actual words, which account for the high UPWR. It is arguable which provides a better user experience. For instance, we can argue that the partial hypothesis provided by the *double decoder* with *please him* is more grammatically correct than *pleasing*. Future studies with different metrics more targeted to partial hypotheses' accuracy and user studies are still needed.

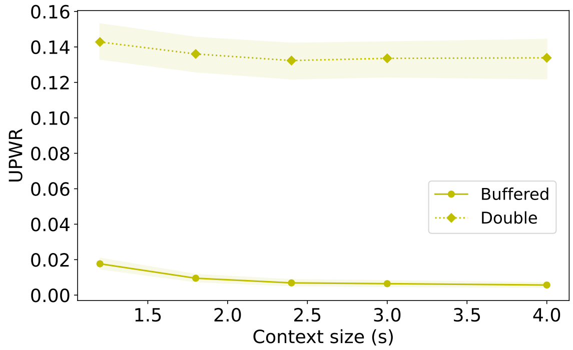


Figure 2: UPWR for LibriSpeech Test-clean, lower is better. The coloured band around the line is the 95% confidence interval.

7.2 Results on Zipformer-CTC-Transducer

As we have shown in the Conformer results that the trend in WER stays the same for Test-clean and Test-other, we feel that Test-clean is sufficient to capture the relevant metrics for the Zipformer demonstration. Table 3 shows the WER results of applying the *double decoder* on the cache-aware Zipformer model. Since this model incorporates a cache for history context, we typically decode with the default method. Comparing with row 1 in Table 2, we can see that this model achieves much better WER even with the default method.

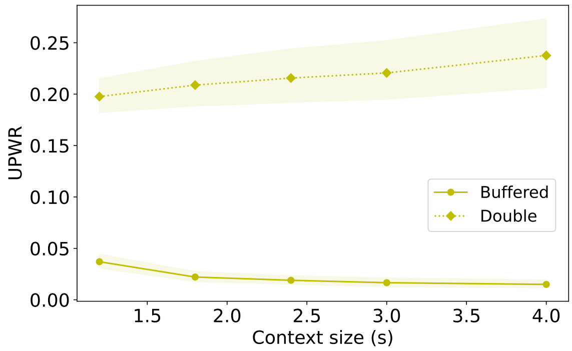


Figure 3: UPWR for LibriSpeech Test-other.

Nonetheless, the results highlights the importance of adding look-ahead context, using either buffered or *double decoding* in different ways. For example, if we compare the results with the same $|x_t|$, by adding 0.6 s of look-ahead, we reduce the WER from 4.45 to 2.98 for cache size of 1.28 s. Similarly, if we compare results given the same context size (see results on the same row), we see WER reduction. Lastly, if we compare row 3 to row 2 and row 6 in Table 3, we can see that for the *double decoder*, decreasing either the look-ahead or history by 0.6 s degrades WER by approximately the same amount, 0.2 for CTC and 0.1 for RNN-T decoding.

Both the buffered and *double decoder* utilizes a look-ahead, but the *double decoder* provides better latency than buffered decoding, given the small $T_d(l_t)$ results in Table 3, making it a better candidate for streaming applications.

Additionally, Table 3 highlights the suitability of *double decoder* for CTC models. We can see that CTC decoding generally provides worse WER, but the improvement from adding context is greater than RNN-T. CTC models also show smaller decoding overhead $T_d(l_t)$.

It should be noted, however, as we are keeping the context sizes constant between default decod-

Table 3: WER (%) on Test-clean of RNN-T greedy decoding for Zipformer-CTC-Transducer for different context sizes. For default decoding, $|x_t|$ is variable while $|l_t| = 0$, for buffered / double decoder, $|x_t|$ is fixed at 0.6 s. We also show $T_d(l_t)$ just as in Table 2.

Context size (s)		Default		Buffered/Double		$T_d(l_t)$ (ms)	
Cache	$\lvert x_t + l_t\rvert$	CTC	RNN-T	CTC	RNN-T	CTC	RNN-T
1.28	0.6	8.43	4.45	N/A	N/A	N/A	N/A
1.28	1.2	6.85	3.91	3.35	2.98	**0.06**	6.75
1.28	1.8	**4.99**	3.04	**3.16**	**2.87**	0.10	10.17
0.64	0.6	8.62	4.57	N/A	N/A	N/A	N/A
0.64	1.2	6.91	3.99	3.45	3.12	0.06	6.75
0.64	1.8	5.06	3.05	3.36	3.00	0.10	10.17

ing and *double decoder*, we reduce the step size $|x_t|$. It means the model is running at smaller intervals, therefore more times for the same duration of audio. The increase in computational cost is non-negligible and should be considered for real-life applications.

8 Conclusions

In this study, we introduce a simple addition to the buffered decoding algorithm, *double decoder* for improving streaming E2E models. Firstly, we show that the use of the *double decoder* improves the appeal of Conformer-CTC models for streaming. With the default or buffered method, it has either unacceptable latency or WER. With the *double decoder*, we reduce latency while maintaining low WER. Secondly, for Zipformer-CTC-Transducer, we show the importance of look-ahead context for further improving the WER. Given the same context size, we are able to achieve better WER and better latency. Given the latency benefits from the *double decoder*, we argue that it is the best method for incorporating look-ahead context. We also explore the side effect of this algorithm, for example, using streaming stability metric UPWR. We observe the degradation in streaming stability by using *double decoder*, and we argue that the context size cannot be too large in real-life noisy conditions. Similarly, we note that extra compute is needed to achieve the result for the Zipformer-CTC-Transducer. For future work, we will investigate whether there are other metrics to fully measure the user perceived readability of partial hypotheses, further improve stability and latency for the studied models, and further investigate the effect of the algorithm when it is applied to different architectures.

Limitations

The main limitation, as we have noted in the main text, is that streaming stability or computational cost worsens when WER or latency improves with our proposed method. Unfortunately the increase in computational cost for the Zipformer is unavoidable, since we are running the encoder at shorter intervals. However, the stability of the partial hypotheses can be further investigated and improved.

Furthermore, we have limited our study to a specific type of E2E models - the Transformer-like models such as the Conformer or the Zipformer. This is due to the fact that Transformer-like models generally achieves better accuracy but have poorer streaming suitability. It can be noted that the proposed method would not provide significant improvement on models such as the LSTM which does not make use of lookaheads or future context, unless it is a bidirectional LSTM. Additionally, we do not have comparisons with other types of fast-slow two-head methods, since their slow head typically involves training a different set of weight and is bound to achieve overall better WER but with more computational cost. We argue that our algorithm is more suitable for smaller-scale research or applications, while it does not achieve the most competitive WER scores.

Lastly, we note that we have used a small set of data for the experiments. Future work is needed to fully evaluate this algorithm on other larger models trained on a variety of datasets.

References

Antoine Bruguier, David Qiu, Trevor Strohman, and Yanzhang He. 2016. Flickering Reduction with Partial Hypothesis Reranking for Streaming ASR. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 38–45. IEEE.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

William Chen, Brian Yan, Jiatong Shi, Yifan Peng, Soumi Maiti, and Shinji Watanabe. 2023. Improving Massively Multilingual ASR with Auxiliary CTC Objectives. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Alex Graves. 2012. Sequence Transduction with Recurrent Neural Networks. In *n International Conference on Machine Learning: Representation Learning Workshop*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, pages 369–376. ACM Press.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech 2020*, pages 5036–5040.

Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, and Alexander Gruenstein. 2019. Streaming End-to-end Speech Recognition for Mobile Devices. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6381–6385.

Fangjun Kuang, Liyong Guo, Wei Kang, Long Lin, Mingshuang Luo, Zengwei Yao, and Daniel Povey. 2022. Pruned RNN-T for fast, memory-efficient ASR training. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2068–2072. ISCA.

Jinyu Li, Rui Zhao, Eric Sun, Jeremy H. M. Wong, Amit Das, Zhong Meng, and Yifan Gong. 2020. High-Accuracy and Low-Latency Speech Recognition with Two-Head Contextual Layer Trajectory LSTM Model. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7699–7703.

Niko Moritz, Takaaki Hori, and Jonathan Le. 2020. Streaming Automatic Speech Recognition with the Transformer Model. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078.

Arun Narayanan, Tara N. Sainath, Ruoming Pang, Jiahui Yu, Chung-Cheng Chiu, Rohit Prabhavalkar, Ehsan Variani, and Trevor Strohman. 2021. Cascaded Encoders for Unifying Streaming and Non-Streaming ASR. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5629–5633.

Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2024. Stateful Conformer with Cache-Based Inference for Streaming Automatic Speech Recognition. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12041–12045.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Golan Pundak and Tara N. Sainath. 2016. Lower Frame Rate Neural Network Acoustic Models. In *Interspeech*, pages 22–26.

Tara N. Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziel Alvarez, Zhifeng Chen, Chung-Cheng Chiu, David Garcia, Alex Gruenstein, Ke Hu, Anjuli Kannan, Qiao Liang, Ian McGraw, Cal Peyser, Rohit Prabhavalkar, Golan Pundak, David Rybach, Yuan Shangguan, Yash Sheth, Trevor Strohman, Mirkó Visontai, Yonghui Wu, Yu Zhang, and Ding Zhao. 2020. A Streaming On-Device End-To-End Model Surpassing Server-Side Conventional Model Quality and Latency. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6059–6063.

Tara N. Sainath, Yanzhang He, Arun Narayanan, Rami Botros, Ruoming Pang, David Rybach, Cyril Allauzen, Ehsan Variani, James Qin, Quoc-Nam Le-The, Shuo-Yiin Chang, Bo Li, Anmol Gulati, Jiahui Yu, Chung-Cheng Chiu, Diamantino Caseiro, Wei Li, Qiao Liang, and Pat Rondon. 2021. An Efficient Streaming Non-Recurrent On-Device End-to-End Model with Improvements to Rare-Word Modeling. In *Proc. Interspeech 2021*, pages 1777–1781.

Tara N. Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, Ian McGraw, and Chung-Cheng Chiu. 2019. Two-Pass End-to-End Speech Recognition. In *Interspeech 2019*, pages 2773–2777.

Yuan Shangguan, Kate Knister, Yanzhang He, Ian McGraw, and Françoise Beaufays. 2020a. Analyzing the Quality and Stability of a Streaming End-to-End On-Device Speech Recognizer. In *Interspeech 2020*, pages 591–595.

Yuan Shangguan, Jian Li, Qiao Liang, Raziel Alvarez, and Ian McGraw. 2020b. Optimizing Speech Recognition For The Edge. In *Third Conference on Machine Learning and Systems, On-Device Intelligence Workshop*.

Yuan Shangguan, Rohit Prabhavalkar, Hang Su, Jay Mahadeokar, Yangyang Shi, Jiatong Zhou, Chunyang Wu, Duc Le, Ozlem Kalinli, Christian Fuegen, and Michael L. Seltzer. 2021. Dissecting User-Perceived Latency of On-Device E2E Speech Recognition. In *Proc. Interspeech 2021*, pages 4553–4557.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *The Twelfth International Conference on Learning Representations*.

Zengwei Yao, Wei Kang, Fangjun Kuang, Liyong Guo, Xiaoyu Yang, Yifan Yang, Long Lin, and Daniel Povey. 2023. Delay-penalized CTC Implemented Based on Finite State Transducer. In *Proc. INTERSPEECH 2023*, pages 1329–1333.

Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N Sainath, Yonghui Wu, and Ruoming Pang. 2021. Dual-mode ASR: Unify and Improve Streaming ASR with Full-context Modeling. In *ICLR 2021*. ICLR 2021.

Xiaohui Zhang, Frank Zhang, Chunxi Liu, Kjell Schubert, Julian Chan, Pradyot Prakash, Jun Liu, Ching-Feng Yeh, Fuchun Peng, Yatharth Saraf, and Geoffrey Zweig. 2021. Benchmarking LF-MMI, CTC And RNN-T Criteria For Streaming ASR. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 46–51.

Personalised Abusive Language Detection Using LLMs and Retrieval-Augmented Generation

Tsungcheng Yao, Ernest Foo**, Sebastian Binnewies***
**School of ICT, Griffith University Nathan Campus, Brisbane, QLD, AU
*School of ICT, Griffith University Gold Coast Campus, Gold Coast, QLD, AU
tsungcheng.yao@griffithuni.edu.au

Abstract

Large language models (LLMs) can be useful tools for detecting abusive language on social media. However, LLMs are not always effective as they can overlook the diversity among individuals, which can lead to severe consequences. This paper proposes a novel solution that incorporates psychological knowledge into an out-of-the-box LLM using the retrieval augmented generation (RAG) method. Two rule sets were extracted and transferred to the LLM via query prompts. Experiment results showed that our solution improves LLM's performance in generating personalised detection by 1.5% to 4.4% weighted F1 score points.

1 Introduction

Abusive language detection systems play a significant role in addressing cyberbullying. Most detection systems function by identifying patterns of abusive messages, such as combinations of letters, keywords, or phrases (Jahan and Oussalah, 2023; Chhabra and Vishwakarma, 2023; Festus Ayetiran and Özgöbek, 2024). In addition, studies have proven that determining abusive language can be greatly impacted by individuals' subjectivity, including attitude, belief and experience (Sap et al., 2022; P.Y.K.L et al., 2024; Wan et al., 2023; Larimore et al., 2021). To make more personalised detection, prior studies have integrated various attributes into the systems, such as Balakrishnan et al. (2020) enhanced detection systems by introducing psychological attributes - Big Five and Dark Triad measurement scales. Kocoń et al. (2021) incorporated user demographic features into their detection systems to make adjusted predictions based on personal profiles.

Leveraging vast training data, LLMs are useful tools for abusive language detection; however, some studies have demonstrated that LLMs are not always effective in detecting such language (Kolla et al., 2024; Kruschwitz and Schmidhuber, 2024). In addition, when dealing with diversity between individuals, Park et al. (2024) found that LLMs can generate near-zero response variation in certain conditions. Overlooking individuals' diversity in abusive detection on LLMs can lead to severe consequences (Cheng et al., 2023; Gallegos et al., 2024). As a result, a novel solution is required to enhance LLMs in generating personalised abusive language detection.

This paper proposes and evaluates a novel solution incorporating psychological knowledge into an LLM (GPT-3.5 Turbo) through the RAG method, initially introduced by Lewis et al. (2020) and later extended for various applications (Fan et al., 2024). Two sets of rules were extracted from a dataset that incorporated psychological features, using association rule mining and a decision tree classifier. Then, these rule sets were provided as extra knowledge to enhance an out-of-the-box LLM's ability to generate personalised detection through the RAG approach. Our experimental results suggest that (i) our solution improves performance and (ii) it is reasonably robust with contradictory inputs. Lastly, the complete code, rules, and data are available on our repository page (here).

2 Method

An experimental approach is adopted to evaluate the effectiveness of the proposed solution (see Figure 1). Two groups, experimental and control, are created. The prompts for the experimental group are enhanced with rules derived from a dataset that includes psychological features, while the prompts for the control group are not enhanced. If the experimental group outperforms the control group in generating personalised detection, we may suggest that the proposed solution is effective and vice versa. The following sections will elaborate on the details of the experiment setup.

Notably, our experiments use a simulator as a preliminary study to assess the feasibility of the

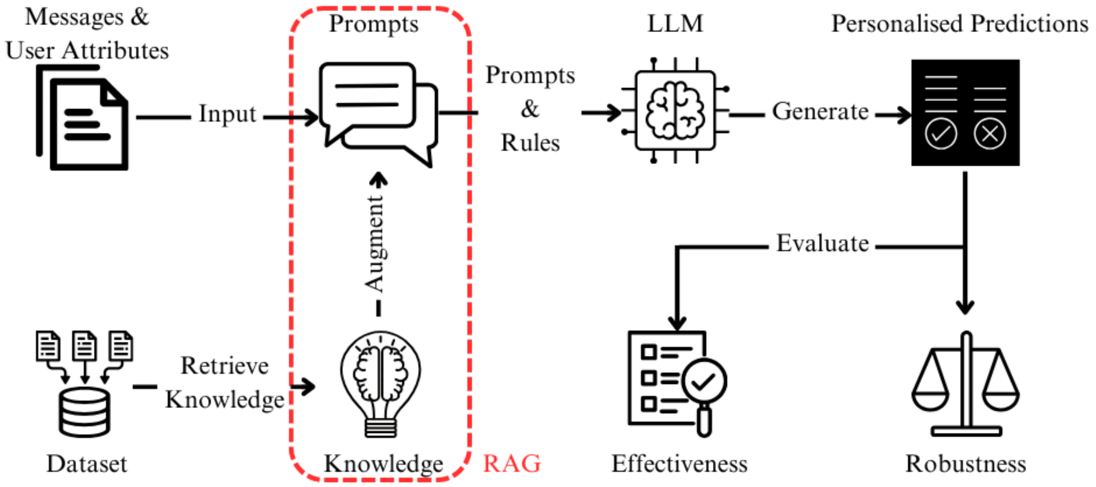


Figure 1: LLM and RAG Personalisation Framework

proposed solution. The input is replaced with test data, and the RAG component is simplified by embedding selected rules into prompts.

2.1 Data

ALDIPF: An Abusive Language Dataset that Includes Psychological Features was used (Yao et al., 2024). ALDIPF denotes 505 users' personality traits and their emotional reactions towards a series of messages with three features: personality traits (user attributes), emotional reactions (class labels), and messages. These user attributes were measured by the Five-Point Shortened General Attitude and Belief Scale, used in clinical settings (Turner et al., 2018). 65.6% of the messages belong to the Neutral class, and the rest are in the Harmful class.

This dataset was created based on the two understandings from psychological studies. First, people's emotional reactions towards messages are co-created by the messages and user attributes. Second, people with similar attributes can share a similar tendency to interpret messages (DiGiuseppe et al., 2013; Ciarrochi and Bailey, 2009). Thus, a single message can be associated with two different class labels, distinguishable only by the users' attributes.

Furthermore, recognising the limitations of LLMs in processing numeric data, user attributes were filtered and transferred into textual tokens. Three attributes, namely Rationality, Irrationality and Self-Down, were selected due to their significance in previous psychological studies (DiGiuseppe et al., 2018; David et al., 2019). After that, the original attributes were transferred from numeric values into 8 buckets (Appendix A) according to the mean and standard deviations (Owings et al., 2013). A data examples are shown in Table 1.

Message	User Attributes	Class
You're SO SMART	Low Ra High SD High Ir	1
You're SO SMART	High Ra Low SD Low Ir	0

Table 1: The same message can be associated with two class labels. Note 1: Class 0 is Neutral, and Class 1 is Harmful. Note 2: Ra indicates Rationality. Ir is Irrationality. SD refers to Self-Down.

2.2 Knowledge Extraction

To extract knowledge from ALDIPF, association rule mining and decision tree approaches were adopted. These processes can establish a correlation between certain user attributes and class labels. In this paper, we are particularly interested in indicators and rules for personalised abusive language detection.

2.2.1 Association Rule Mining Approach

Association rule mining can discover items' co-occurrence probability by identifying frequently occurring item sets and generating rules among them. This approach has been proven effective in extracting rules in various study settings (Diaz-Garcia et al., 2023; Shu and Ye, 2023).

Nine apriori algorithm-generated rules were selected after evaluating their support, confidence and lift. Five were associated with the Harmful class

Item Set	Conseq	Conf	Lift
Low Ra Ex High SD	Harmful	0.78	2.31
Very High Ir Low Ra	Harmful	0.72	2.13
Very High Ir Low Ra Ex High SD	Harmful	0.78	2.31
Low Ra, Ex High SD	Harmful	0.78	19
Very High Ir Very High SD Low Ra	Harmful	0.66	1.95

Table 2: Rules based on association rule mining approach

(see Table 2), and the rest belonged to the Neutral class.

Condition	Class	Prob
In: Below Avg. Ir Ex High SD Not in: Higher Ra	Harmful	0.561
In: Very High SD Below Avg. Ra	Harmful	0.798
In: Very High Ra Not in: Below Avg. SD Very High Ir	Harmful	0.583
In: Low SD Low Ir Not in: High Ra	Harmful	0.861

Table 3: Rules based on Decision Tree. Note: these rules have been simplified for better readability.

2.2.2 Decision Tree Approach

Decision trees are widely used for knowledge representation due to their high interpretability (Shu and Ye, 2023; Sarailidis et al., 2023). This approach creates a tree-like model of decisions by splitting data into subsets and repetitive recursion.

The decision tree classifier algorithm generated rules, thirteen of which were selected after evaluating class distribution and probabilities. Four were associated with the Harmful class (see Table 3), and the rest belonged to the Neutral class. A complete decision tree is available on our GitHub page.

2.3 LLM and Prompt Optimization

The experiments were conducted on GPT-3.5 Turbo via the OpenAI API with a temperature setting of 0. Our prompts followed the framework proposed by Eager and Brunton (2023). Additionally, we used the attributed prompt technique to co-create prompts with the LLM, improving performance, particularly in the transformation of rules (Yu et al., 2023). The complete prompt is provided in Appendix B, and the design of the prompts is as follows:

- Role of the LLM: It was assigned as a classifying bot that helps users classify messages.

- The task with details: The LLM was asked to analyze the following messages and determine whether it was abusive. A message is considered abusive if it might make a reader uncomfortable after reading it. In other words, we focus on not only explicit but also implicit abusive messages.

- Additional Knowledge: The LLM was informed that the decision could be made based on the textual message and user attributes. Then, the rules discussed in Section 2.2 were presented. In our experiment, selected rules were embedded into the prompts.

- Input: Consists of (i) a message and (ii) user attributes.

- Output: A score from 0 to 1, where 0 means absolutely not harmful, and 1 means definitely harmful.

Three prompt architectures were created by modifying the *Additional Knowledge* layer. DT_M denotes the architectures enhanced by the decision tree rules. ARM_M is enriched by association rule mining. Lastly, there is no augment for N_M, and the user attributes were removed from the *Input* layer.

3 Experiment and Result

Two experiments were conducted to evaluate the effectiveness and robustness of the proposed solu-

Archt.	ACC	Weighted F1	TP Rate
N_M	0.600	0.556	0.192
DT_M	**0.630**	**0.600**	**0.258**
ARM_M	0.608	0.571	0.225

Table 4: Result for Experiment 1. Note: TP rate is defined as Number of True Positives / Number of Positive Samples

tion.

3.1 Experiment 1: Effectiveness

This experiment evaluates whether the proposed approach can enhance the LLM's ability to generate personalised detection.

Implementation: 500 data points containing messages and user attributes were randomly selected from ALDIPF, and 36% were Harmful class. Then, these selected data points were conveyed into the *Input* layer of the prompt framework. Importantly, messages and user attributes were passed to DT_M and ARM_M, while only messages were passed to N_M. After that, the LLM's responses were cleaned and rounded to 0 or 1. Lastly, all responses were evaluated against the ordinary class labels.

Result: The experimental group consistently outperformed the control group at every metric (see Table 4). Importantly, the experimental group yielded higher true positive rates (TP rate), which implies that the experimental group can identify more abusive messages than its counterpart.

3.2 Experiment 2: Robustness

Prior studies suggested that individual subjectivity should be less influential in determining abusive messages when the messages usually have only one clear meaning (Sandri et al., 2023; Plank, 2022). Therefore, this experiment assesses whether the proposed solution can handle contradictory inputs, such as extremely positive or negative messages paired with attributes that strongly contrast the messages.

Implementation: 100 joyful messages were randomly selected from the HappyDB (Asai et al., 2018), a corpus of 100,000 happy moments. Then, these messages were joined with attributes strongly related to the Harmful class (Appendix C).

Regarding profane messages, ChatGPT created 100 samples containing at least one swear word. Then, these samples were joined with attributes strongly related to the Neutral class (Appendix C).

Type	N_M	DT_M	ARM_M
Joyful	[100,0]	[98,2]	[97,3]
Profane	[0,100]	[1,99]	[**9,91**]

Table 5: Result for Experiment 2. Note: [Neutral class, Harmful class]

Similar to experiment one, both messages and attributes were passed to DT_M and ARM_M, while only messages were passed to N_M. As a result, N_M was not impacted by manipulated attributes. In this instance, N_M serves as the baseline to evaluate the extent to which the proposed solution would be affected by contradictory inputs.

Result: For N_M, both joyful and profanity messages were accurately classified according to their nature. Nevertheless, the experimental group encountered different levels of disturbance (see Table 5). In particular, the predictions in Profanity were flipped by 9% in ARM_M.

4 Discussion

Comparison of Rules. Although the knowledge extraction approaches differ, the two rule sets still share similarities. Extremely High Self-Down is always associated with the Harmful class. In addition, Low and Lower Rationality are generally linked with the Harmful class. Nevertheless, the correlation between the Irrationality and Harmful class is unclear due to the contradictory implications of the two rules. Specifically, association rule mining indicates a positive correlation between Irrationality and the Harmful class, whereas decision tree analysis shows the opposite.

Effectiveness of Solution. The experimental group showed consistent improvement on every metric. Importantly, part of the improvement stems from identifying more abusive language (higher TP rate). As a result, the experimental group can provide more benefits for users, as identifying abusive language is the primary advantage users gain from detection systems (Hardt et al., 2016).

Robustness of Solution. Despite extremely joyful or profane messages, the results of Experiment 2 did not align with the expectation that individual subjectivity would be less influential when messages have a clear meaning.

Regarding the joyful message, our solution is robust to noise, effectively addressing contradictory inputs. However, more research is needed on profane messages. When explicit swear words were presented, DT_M predictions were flipped by only 1%, while ARM_M predictions were flipped by 9%.

5 Conclusion and Future Work

This paper evaluates a novel solution that enhances LLMs for personalised abusive language detection by retrieving and incorporating psychological knowledge into an out-of-the-box LLM. Experiment results showed that our solution outperformed its counterpart and withstood noise reasonably well. For better re-productivity, the complete code, rules, and data are available on our repository page (here).

In our future work, we aim to address two key limitations. First, more detailed decision rules will be developed to cover a broader range of user attributes, making the retrieval component essential and requiring additional evaluation of its effectiveness. Second, further research should explore the compound effects between messages and user attributes. While this paper focuses on testing the robustness of the proposed solution in highly contradictory scenarios, the major challenge of robustness, in our view, lies in handling neutral messages across diverse individual attributes.

References

Akari Asai, Sara Evensen, Behzad Golshan, Alon Halevy, Vivian Li, Andrei Lopatenko, Daniela Stepanov, Yoshihiko Suhara, Wang-Chiew Tan, and Yinzhan Xu. 2018. HappyDB: A corpus of 100,000 crowdsourced happy moments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Vimala Balakrishnan, Shahzaib Khan, and Hamid R. Arabnia. 2020. Improving cyberbullying detection using twitter users' psychological features and machine learning. *Computers & Security*, 90:101710.

Myra Cheng, Esin Durmus, and Dan Jurafsky. 2023. Marked personas: Using natural language prompts to measure stereotypes in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1504–1532, Toronto, Canada. Association for Computational Linguistics.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, 29(3):1203–1230.

Joseph Ciarrochi and A. Bailey. 2009. *A CBT-practitioner's Guide to ACT: How to Bridge the Gap between Cognitive Behavioral Therapy and Acceptance and Commitment Therapy*, volume 50.

Daniel O. David, Raymond DiGiuseppe, Anca Dobrean, Costina Ruxandra Păsărelu, and Robert Balazsi. 2019. *The Measurement of Irrationality and Rationality*, pages 79–100. Springer International Publishing, Cham.

J.A. Diaz-Garcia, M.D. Ruiz, and M.J. Martin-Bautista. 2023. A survey on the use of association rules mining techniques in textual social media. *Artificial Intelligence Review*, 56:1175–1200.

Raymond DiGiuseppe, Russell Leaf, Bernard Gorman, and Mitchell W. Robin. 2018. The development of a measure of irrational/rational beliefs. *Journal of Rational-Emotive & Cognitive-Behavior Therapy*, 36(1):47–79.

Raymond A. DiGiuseppe, Kristene A. Doyle, Windy Dryden, and Wouter Backx. 2013. *A Practitioner's Guide to Rational-Emotive Behavior Therapy*. Oxford University Press.

B. Eager and R. Brunton. 2023. Prompting higher education towards ai-augmented teaching and learning practice. *Journal of University Teaching & Learning Practice*, 20(5).

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Eniafe Festus Ayetiran and Özlem Özgöbek. 2024. A review of deep learning techniques for multimodal fake news and harmful languages detection. *IEEE Access*, 12:76133–76153.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics*, pages 1–79.

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA. Curran Associates Inc.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, page 126232.

Jan Kocoń, Alicja Figas, Marcin Gruza, Daria Puchalska, Tomasz Kajdanowicz, and Przemysław Kazienko. 2021. Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing & Management*, 58(5):102643.

Mahi Kolla, Siddharth Salunkhe, Eshwar Chandrasekharan, and Koustuv Saha. 2024. Llm-mod: Can large language models assist content moderation? In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Udo Kruschwitz and Maximilian Schmidhuber. 2024. LLM-based synthetic datasets: Applications and limitations in toxicity detection. In *Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying @ LREC-COLING-2024*, pages 37–51, Torino, Italia. ELRA and ICCL.

Savannah Larimore, Ian Kennedy, Breon Haskett, and Alina Arseniev-Koehler. 2021. Reconsidering annotator disagreement about racist language: Noise or signal? In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 81–90, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Larry R. Owings, Gregory L. Thorpe, Evan S. McMillan, Ronald D. Burrows, Scott T. Sigmon, and Dawn C. Alley. 2013. Scaling irrational beliefs in the general attitude and belief scale: An analysis using item response theory methodology. *SAGE Open*, 3(2).

P.S. Park, P. Schoenegger, and C. Zhu. 2024. Diminished diversity-of-thought in a standard large language model. *Behavior Research Methods*.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Srinivas P.Y.K.L, Amitava Das, and Viswanath Pulabaigari. 2024. Racists spreader is narcissistic; sexists is machiavellian influence of psycho-sociological facets in hate-speech diffusion prediction. *Expert Systems with Applications*, 247:123211.

Marta Sandri, Elisa Leonardelli, Sara Tonelli, and Elisabetta Jezek. 2023. Why don't you do it right? analysing annotators' disagreement in subjective tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441, Dubrovnik, Croatia. Association for Computational Linguistics.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Georgios Sarailidis, Thorsten Wagener, and Francesca Pianosi. 2023. Integrating scientific knowledge into machine learning using interactive decision trees. *Computers Geosciences*, 170:105248.

Xiaoling Shu and Yiwan Ye. 2023. Knowledge discovery: Methods from data mining and machine learning. *Social Science Research*, 110:102817.

Martin J. Turner, Mark S. Allen, Matthew J. Slater, Jamie B. Barker, Charlotte Woodcock, Chris G. Harwood, and Ken McFayden. 2018. The development and initial validation of the irrational performance beliefs inventory (ipbi). *European Journal of Psychological Assessment*, 34(3):174–180.

Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. 2023. Everyone's voice matters: Quantifying annotation disagreement using demographic information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14523–14530.

Tsungcheng Yao, Sebastian Binnewies, Ernest Foo, and Masoumeh Alavi. 2024. See the words through my eyes: The role of personality traits in abusive language detection. *SSRN Electronic Journal*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Buckets

The numeric psychological features were transferred to 8 buckets as shown in Table 6.

Buckets	Description
Extremely High	Any $> (\overline{X} + 3\sigma_X)$
Very High	$(\overline{X} + 2\sigma_X)$ to $(\overline{X} + 3\sigma_X)$
High	$(\overline{X} + \sigma_X)$ to $(\overline{X} + 2\sigma_X)$
Above Average	$\overline{X}$ to $(\overline{X} + \sigma_X)$
Below Average	$\overline{X}$ to $(\overline{X} - \sigma_X)$
Low	$(\overline{X} - \sigma_X)$ to $(\overline{X} - 2\sigma_X)$
Very Low	$(\overline{X} - 2\sigma_X)$ to $(\overline{X} - 3\sigma_X)$
Extremely Low	Any $< (\overline{X} - 3\sigma_X)$

Table 6: Buckets and Ranges of Values

B Prompt Details

System Role: You are a classifying bot that helps users classify sentences. Please analyze the following sentence and determine if it is abusive or not. A sentence is considered abusive if a reader may feel uncomfortable after reading it. Provide a score from 0 to 1, where 0 means absolutely not harmful, and 1 means definitely harmful. Lastly, just return the score and do not provide any explanation. Additional Knowledge Base:
Here are the association rules mined from user attributes and their impact on identifying harmful messages:

Rule 1:
Antecedents: Low Rationality, Ex High SD
Consequents: discomfort
Support: 0.0014
Confidence: 0.78
Lift: 2.31
Explanation: When users exhibit low rationality and extremely high standard deviation in identity, there is a strong association with message content leading to discomfort (discomfort), with a lift of 2.31.

Rule 2:
Antecedents: Very High Irrationality, Low Rationality
Consequents: discomfort
Support: 0.0025
Confidence: 0.72
Lift: 2.13
Explanation: The combination of very high irrationality and low rationality significantly correlates with messages causing discomfort (discomfort), with a lift of 2.13.

Rule 3:
...
...
note: complete code and rules are available on our repository page (here).

User Role: Sentence: this is a sample message. Attributes: this is a sample attribute.

C User Attributes for Experiment 2

For ARM_M, joyful messages were attached to Low Rationality and Extremely High Self-Down, strongly associated with Harmful class, while profane messages were linked to Low Self-Down, strongly associated with Neutral class.

For DT_M, joyful messages were attached to Very High Self-Down and Low Rationality, strongly associated with Harmful class, while profane messages were linked to Below Average Self-Down and High Rationality, strongly associated with Neutral class.

Large-scale Summarization of Chat Transcripts in the Absence of Annotated Summaries

Pratik K. Biswas
Artificial Intelligence and Data, Verizon Communications
Basking Ridge, New Jersey, USA 07920
pratik.biswas@verizonwireless.com

Abstract

Text summarization is the process of condensing a piece of text to fewer sentences, while still preserving its content. Chat transcript, in this context, is a textual copy of a digital/web conversation between a customer (caller) and agent(s). This paper presents a locally developed hybrid method that combines extractive (unsupervised) and abstractive (supervised) summarization techniques in compressing ill- or un-punctuated chat transcripts to produce more readable summaries. Extensive testing, evaluation and comparisons have demonstrated the efficacy of this approach, in the absence of annotated (reference) summaries, for large-scale summarization.

1 Introduction

Automatic document summarization aims to compress a textual document to a shorter, more informative format while keeping key information of the original text. Numerous approaches have been developed for automatic text summarization and can be broadly classified into two groups: extractive and abstractive summarization. Extractive summarization extracts important sentences from the original text and reproduces them verbatim in the summary, while abstractive summarization generates new sentences. Hybrid Summarization attempts to combine these two approaches in some form.

Chat transcription is defined as the process of converting a digital or web conversation into written words to be stored as plain text in a conversational language. In this paper, however, we will be confining ourselves to *textual descriptions of web chats between customer (caller) and agent(s) (customer representatives) of a* ***phone company***. Automatic summarization of chat transcripts, in this context, pose certain unique challenges, as follows: 1) they are not continuous texts but include conversations between customers and agents, 2) they are often very short or very long, and can include a large number of sentences that are irrelevant and even meaningless, 3) they include several ill-formed, grammatically incorrect sentences, 4) they are either un- or improperly punctuated, 5) there is a dearth of a large collection of human-crafted annotated (reference) summaries that can be used as training samples and 6) existing open-source summarization tools don't perform well with chat transcripts unless properly customized or fine-tuned.

In this paper, we have presented a hybrid summarization technique that combines extractive summarization, comprising of *channel separation* (separation into customer and agent transcripts), *topic modeling, sentence selection* and *punctuation restoration* with *supervised* abstractive summarization via *transfer learning* to produce properly punctuated, fixed-length and readable customer and agent summaries, from the original chat transcripts, that can adequately summarize customer concerns and agent resolutions.

2 Related Work

Related research can be broadly grouped into three categories: 1) extractive, 2) abstractive and 3) hybrid Summarization.

Radev et al. (2002) defined summary as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that." Automatic text summarization gained attraction as early as the 1950s. Different methods and extensive surveys of automatic text summarization have been provided in (Zechner, 1997; Mani, 2001; Jones, 2007; Jezek and Steinberger, 2008; Nenkova and McKeown, 2012; Saggion and Poibeau, 2013).

Luhn (1958) introduced a method to extract salient sentences from the text using features such as word and phrase frequency. Gong and Liu (2001) and Wang et al. (2009) summarized multiple documents using topic models. Miller (2019) used Bidirectional Encoder Representations from Transformers (BERT) for summarization of lecture notes. Liu (2019) described BERTSUM, a simple variant of BERT, for extractive summarization. Liu and Lapata (2019) showcased how BERT could be generally applied in text summarization and proposed a general framework for extractive and abstractive models. Feigenblat et al. (2021) introduced TWEETSUM, a large-scale database of customer support dialogs with extractive and abstractive summaries along with an unsupervised extractive summarization method, specific to these dialogs.

Lin and Ng (2009) and Khan and Salim (2014) reviewed the various methods for abstractive summarization. Nallapati et al. (2016), Paulus et al. (2017), See et al. (2017) and Liu et al. (2017) employed recurrent neural networks, deep reinforcement learning, pointer-generator and generative adversarial networks for abstractive summarization. Lewis et al. (2019) introduced BART, a denoising autoencoder for pre-training sequence-to-sequence models that was particularly effective when fine-tuned for text generation (e.g., abstractive summarization, translation, etc.). Beltazi et al. (2020) presented Longformer, useful for long document summarization. Tuggener et al. (2021) provided an extensive overview of existing dialog summarization data sets and mappings from data sets to linguistic models. Fabbri et al. (2021) crowdsourced four new datasets from news comments, discussion forums, community question answering forums, as well as email threads and benchmarked state-of-the-art abstractive summarization models on their datasets. Zhong et al. (2021) presented DialogLM, a pre-trained neural encoder-decoder model for long dialog understanding and abstractive summarization.

Bae et al. (2019) followed a hybrid architecture, rewrote sentences from a document and then paraphrased the selected ones to generate a summary. Su et al. (2020) combined the two summarization methods to generate a variable-length, fluent summary.

3 Major Contributions

Our main contributions and advantages can be summarized as follows:

1. We have integrated topic modeling and embedding based sentence selection with transformer (BERT) based punctuation restoration for extractive summarization through a 10-step sequential procedure.

2. We restore punctuation in the summaries of un-punctuated or ill-punctuated transcripts.

3. We have fine-tuned powerful, transformer-based language models, on locally extracted summaries, for abstractive summarization of chat transcripts through transfer learning.

The summaries can be useful both as historical records and reminder messages of prior chats.

4 Hybrid Summarization of Chat Transcripts

We propose a hybrid strategy that includes extraction, fine-tuning, and abstraction. Its main objective is to provide a hybrid summarization framework that can first extract the summaries of transcripts to create a large enough training sample, and then use this sample to fine-tune pre-trained language model based abstractive summarizers to generate new summaries of unseen transcripts through transfer learning. The resultant summaries are expected to be at least as good as the extractive summaries, with the tacit expectation that the pre-training encoded in the abstractive approach would make the summaries even more fluent, coherent and help reduce some grammatical errors found in the original transcripts. So, the strategy involves 2 sequential phases. Phase I uses an extractive summarizer, while Phase II uses abstractive ones. The abstractive summarizers depend upon extractive summarizer's outputs for their fine-tuning (supervision). The strategy is useful in a production environment which requires the summarization of a very large number of chat transcripts but where there is a paucity of *manually generated* reference (annotated) summaries from which the abstractive summarizers can learn in Phase II and with which we can compare our results. Figure 1 shows the two Phases of the proposed hybrid summarization strategy.

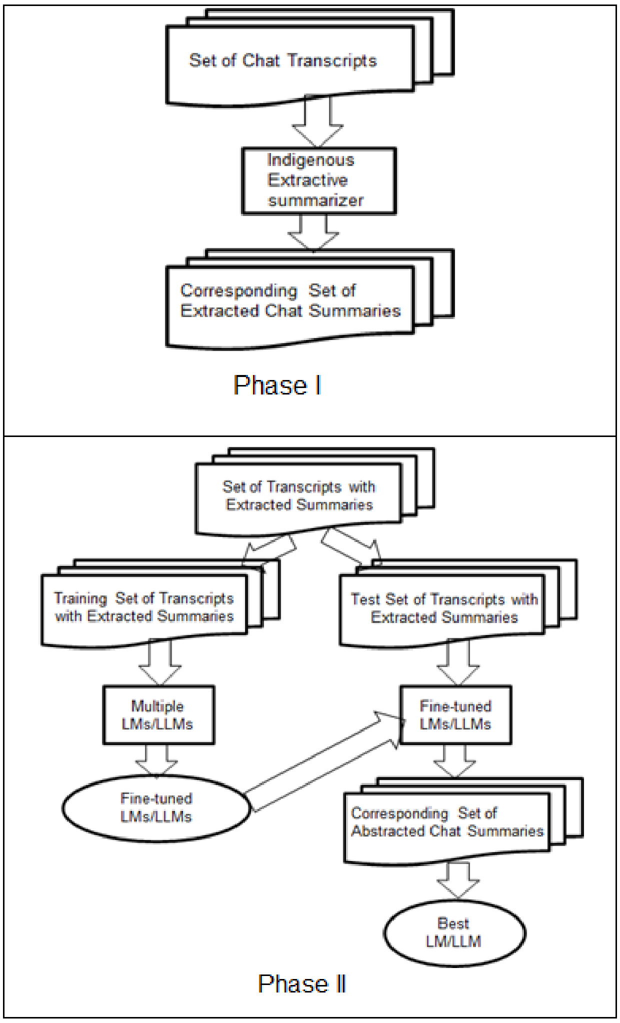

Set of Chat Transcripts
Indigenous Extractive summarizer
Corresponding Set of Extracted Chat Summaries
Phase I
Set of Transcripts with Extracted Summaries
Training Set of Transcripts with Extracted Summaries
Test Set of Transcripts with Extracted Summaries
Multiple LMs/LLMs
Fine-tuned LMs/LLMs
Fine-tuned LMs/LLMs
Corresponding Set of Abstracted Chat Summaries
Best LM/LLM
Phase II

Figure 1: Hybrid Summarization.

5 Phase I: Sample Generation through Extractive Summarization

Phase I generates a large pool of chat summaries, through extractive summarization, which can be reused for fine-tuning supervised abstractive summarizers. This extractive summarization technique uniquely integrates *channel (speaker) separation*, *topic modeling*, and *similarity-based sentence selection* with *punctuation restoration* through a 10-step sequential procedure. It is *internally developed* based on an adaptation from (Biswas and Iakubovich, 2022).

The *punctuation restored summaries* are the outputs from this procedure. The procedure is highly parameterized. The full list of parameters to the proposed procedure includes: *Topic Model Type* (default: "None/False"), *Number of Topics* (default: 5), *Number of Dominant Topics* (default: 1), *Batch Size for Punctuation Restoration* (default: 512), *Term Extraction Method* (default: "global"), *Desired Summary Length* (default: 5), *Summary Table Name* (default: "summary_results"), *Word Similarity Threshold* (default: 0.5), *Uniqueness Threshold for Sentence Similarity* (default: 0.5). Next, we describe the key steps of this procedure.

5.1 Channel Separation

Chat transcripts include conversations/dialogs between customer and one or more agents and so the resultant summaries can often get mixed up. The separation of a transcript into customer and agent transcripts, based on channel or speaker identifier, can make each summary more coherent.

If the channel identifiers, associated with the transcripts, do not clearly identify the speakers then we can use a *pre-trained BERT Transformer* model with a *linear classifier* from *PyTorch nn* package as an additional layer, on top of BERT's 12 layers, to classify each dialog of the transcript into one of the two classes, i.e., *customer* and *agent* and then combine each type of dialogs to create customer and agent transcripts. We haven't used this with our chat transcripts as the speakers were identified.

5.2 Document Preparation

A *document* is a list of *keywords* extracted from each transcript and is used as input to the topic model. For document preparation, we have built a custom NLP preprocessing pipeline comprising of tokenization; removal of punctuation, extended stop-words and small words (length ≤ 4); regular expression matching; lowercasing; contraction mapping; bigrams and trigrams creation; lemmatization; parts of speech tagging and allowable tag selection. This has been implemented by combining modules available from four *Python* packages, namely, *re*, *spaCy*, *NLTK,* and *gensim*.

5.3 Topic Model Optimization and Optimal Model Selection

If the *topic model type* is specified at the invocation of the procedure, then we create multiple *topic models* (instances) of the desired type, for both customer and agent, using the *documents*, *corpus* and *vocabulary* from the corresponding chat transcripts, by varying the hyper-parameter (e.g., *topic number*) values within the pre-defined ranges by the pre-defined steps; compute their coherence scores and identify the topic models and associated hyper-parameter values that produce the best scores. Otherwise, by default, we perform the above-mentioned activity for all 3 different topic model types, namely, *LDA*, *LSI* and *HDP*, in parallel, and identify the topic models and associated hyper-parameter values that produce the best scores amongst topic models of all three types, through an extensive grid search over a wide range of values. We have exclusively used the *Python* based *gensim* package for this step.

5.4 Punctuation Restoration

The punctuation restoration algorithm is used in steps 2 and 8 of the aforesaid procedure. In step 2, we preprocess transcripts (customer and agent) to remove existing punctuations and then restore punctuations *partially*, i.e., restore only *periods* as delimiters, so that sentences can be separated in each transcript; while in step 8, we remove *existing periods* from each pair of customer and agent summaries, restore *partial* and *full punctuations* and postprocess them for more readable outputs.

We have used the *BertForMaskedLM* class of the *PyTorch* BERT model (*bert-base-uncased*[1]) for punctuation restoration and added an additional *linear* layer (*PyTorch nn module*) above the 12 BERT layers. The output of original BERT layers is a vector with the size of all vocabulary. The additional linear layer takes this as input and gives as output one of four classes, i.e., “O” (Other), “Comma”, “Period” and “Question” for each encoded word. We retrained this modified BERT model using *TED transcripts*, consisting of two million words. This retraining with the proposed architecture is unique for punctuation restoration.

We found that the BERT model for punctuation restoration gave **30%** more accurate results than the LSTM based model. We implemented the punctuation restoration algorithm using *BERT Transformer, BertPunc* and *nn* packages, available from *PyTorch*.

5.5 Summary Generation through Sentence Selection

This process combines steps 5 through 7 of the main procedure, i.e., dominant topic identification, significant term selection and summary generation. First, we get the most *dominant topic(s)* from the selected topic models (for customer and agent) with the associated keywords for each of customer and agent documents for every transcript. Second, we use the keywords/terms associated with customer and agent dominant topics to extract the most significant inter-related *terms* for each transcript (document) pair using *word-based similarity analysis* and construct a string/document with them. Lastly, we generate fixed-length customer and agent summaries for every chat transcript, using *embedding-driven*, *sentence-based similarity analysis*. First, we condense each of customer and agent transcripts by identifying its most unique sentences and then we select a fixed number (user-specified) of most relevant sentences from the condensed transcripts that are *most similar* to the string/document constructed at the previous step.

For *term-based similarity analysis*, we have calculated *cosine similarity* between *GloVe* encoded word vectors (300 dimensions) using *spaCy*’s *en_vectors_web_lg*; while for *sentence-based similarity analysis and summary generation*, we have used the *Universal Sentence Encoder (USE)* from *tensorflow-hub,* along with the *Python* based *pandas* and *numpy* packages.

5.6 Summarization Evaluation

We have determined the effectiveness of the summarizer by measuring both the *goodness* (*quality*) of summarization and the *correctness* (*accuracy*) of the punctuation restoration reflecting the content and readability of the summaries.

For the goodness/*quality* of the information content of the generated summaries, the metrics *BLEU* [Bilingual Evaluation Understudy] (Papineni et al., 2002) and *ROUGE* [Recall-Oriented Understudy for Gisting Evaluation] (Lin, 2004) scores can be used as measurements. We have computed the *BLEU and ROUGE-l* scores using the *Python* packages *NLTK* (*nltk.translate.BLEU_score*) and *ROUGE* (*rouge*).

---

[1] `https://huggingface.co/google-bert/bert-base-uncased`

For the *correctness/accuracy* of the punctuation restoration, the *accuracy_score* function from python's *sklearn.metrics* package can measure the *punctuation-restoration-accuracy* (Biswas and Iakubovich, 2022), as the number of matches of punctuation symbols (periods) between the original/extracted text (transcript/summary) and the punctuated text (transcript/summary), expressed as a percentage.

6 Phase II: Abstractive Summarization through Transfer Learning

Phase II involves fine-tuning pre-trained *Large Language Model* (*LM/LLM*) driven, transformer based *abstractive summarizers* on the extractive summaries, obtained from Phase I, and then using them in generating summaries from unseen chat transcripts, via *transfer learning*, to find the most effective *fine-tuned* summarizer for potential production deployment. We have fine-tuned four pre-trained LM/LLM driven transformers, i.e., **T5** (*t5-small* [2]), **BART** (*bart-large-xsum* [3]), **Longformer2Roberta** (*longformer2roberta-cnn_dailymail-fp16* [4]), and **DialogLED** (*DialogLED-large-5120*[5]) for this purpose. **T5** (Raffel et al., 2019), an encoder-decoder model, was pre-trained on Common Crawl and encodes at-most 512 tokens. **BART** is a denoising transformer encoder-decoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder, fine-tuned on the Extreme Summarization (XSum) dataset. It encodes up to 512 tokens. **Longformer2Roberta** is an encoder-decoder model, where the encoder is an *allenai/longformer-base-4096* model and the decoder is a *roberta-base* model, fine-tuned on the CNN/DailyMail dataset. It can handle up to 4096 tokens. **DialogLED** is a pre-trained model for long dialog understanding and summarization. It builds on the Longformer-Encoder-Decoder architecture and uses window-based denoising as the pre-training task on a large amount of long dialog data, encoding up to 5120 tokens. We selected **T5** as it was then the state-of-the-art, **BART** as it was then commonly used for dialog summarization, **Longformer2Roberta** as some chat transcripts were long documents and **DialogLED** as it was designed to improve dialog summarization.

7 Performance Evaluation

Effectiveness (quality of summaries), efficiency (summarization/fine-tuning time), flexibility and performance comparisons with/among open-source, off-the-shelf summarizers are some of the considerations that helped us evaluate the performances of our strategy for chat transcript summarization.

7.1 Experimental Setup

We set up a Spark cluster, consisting of a driver node and dynamically allocated, multiple executor nodes for data collection, preprocessing and summarization. The NVIDIA CUDA Deep Neural Network (cuDNN v7.6) accelerated our training process for punctuation restoration. We retrained, fine-tuned, and tested the transformer models on NVIDIA Tesla A100-SXM4-40GB GPU based nodes, using anywhere between 1 to 4 GPUs.

In Phase I, we tested our extractive summarizer on a ***dataset*** consisting of **160,000** chat transcripts, covering a wide range of issues including *billing, refunds, upgrades, service, outage, maintenance,* etc. The average and maximum lengths of the full chat and the constituent customer and agent transcripts were (314, 7295), (92, 4225) and (222, 4064) words respectively. We compared the performances of our summarizer with those from another very popular, open-source extractive summarizer, namely, *BERT Extractive Summarizer*[6] using three pre-trained transformer models: **BERT** (*bert-base-uncased*[1]) [encoder], **GPT-2** (*gpt2-medium*[7]) [decoder], and **XLNet** (*xlnet-base-cased*[8]) [decoder]. We chose these three models as they could summarize well without

---

[2] `https://huggingface.co/google-t5/t5-small`

[3] `https://huggingface.co/facebook/bart-large-xsum`

[4] `https://huggingface.co/patrickvonplaten/longformer2roberta-cnn_dailymail-fp16`

[5] `https://huggingface.co/MingZhong/DialogLED-large-5120`

[6] `https://pypi.org/project/bert-extractive-summarizer/`

[7] `https://huggingface.co/openai-community/gpt2-medium`

[8] `https://huggingface.co/xlnet/xlnet-base-cased`

fine-tuning. *BERT Extractive Summarizer* generated summaries using the *period-restored* chat, customer, and agent transcripts from step 2 of the proposed extractive summarization procedure. Its *ratio* parameter was adjusted, using the number of words in the transcript, to ensure that its summaries were of comparable (shorter) lengths. The transcripts were summarized both *with* and *without channel separation (full chat),* as part of an ***ablation study***.

In Phase II, the **160K** transcripts (customer & agent) with their corresponding extractive summaries (from Phase I) were split into 3 sets, i.e., *train*, *test* and *hold-out*, with **150K**, **5K** and **5K** samples respectively. We fine-tuned the pre-trained **T5**, **BART**, **Longformer2Roberta** and **DialogLED** models on train and test sets for abstractive summarization and validated their summaries on the hold-out set. The hyper-parameters of the models, e.g., *ecncoder_length*, *decoder_length, batch_size, num_beams*, *learning_rate*, *weight_decay*, *num_train_epochs, fp16,* etc., were tuned to generate better summaries.

The open-source summarizers were used with their respective pre-trained models, *tokenizers*, *configurations*, *vocabularies,* and *checkpoints*.

7.2 Manual Evaluation

The summaries generated by the proposed method are being manually validated for content and readability by our business customers. The goal is to subjectively evaluate if the summaries can be deemed generally useful for the very purposes that the transcripts were meant to be used. Feedback includes the following.

- For ~50 or so chat transcripts, our extractive summaries aptly matched manual summaries.

- The abstractive summaries were readable, generally comparable to the extractive summaries and mostly expressed the main information content of the original transcripts.

- If the chat was about one problem, then ~80% of the transcripts were capably summarized.

- Punctuations greatly improved the readability of the generated summaries.

- Our extractive summaries were more meaningful and readable than the summaries generated by their existing methods, namely, *genism summarizer*, *pytextrank*, *pysummarization auto-abstractor* for their use cases.

- The abstractive summaries didn't include opinions outside of those expressed in the extractive summaries (absence of bias).

- The abstractive summaries generated by **BART** matched the extractive summaries more than **T5**, **Longformer2Roberta** and **DialogLED**.

7.3 Automatic Evaluation

We evaluated our summarizer for *effectiveness* and *efficiency*. For measuring the *effectiveness* of our summarization and for comparing performances among extractive and abstractive summarizers, we have used the metrics *BLEU* and *ROUGE-l scores* (Sec. 5.6). We determined the *efficacy* of our *punctuation restoration* algorithm in Phase I using *punctuation-restoration-accuracy score* (Sec. 5.6).

The *efficiency* of a summarizer is important to real world applications. For Phase I, we have measured the *efficiency* of our extractive summarizer by recording the time taken by each of the 10 steps of our proposed procedure. We have also compared the *efficiency* of our *summary generation* process (Sec. 5.5) with that of the *BERT Extractive Summarizer* by comparing the *total time* taken by each to *summarize* all of chat, customer and agent transcripts in the 160K sample. For Phase II, we have compared the efficiency of the four abstractive summarizers by their *average fine-tuning times* on customer and agent transcripts.

7.4 Results and Summarizer Comparisons

Table 1 shows results from Phase I and compares the *effectiveness* & *efficiency* of the proposed summarizer for shorter summaries (~5 sentences) with those from the *BERT Extractive Summarizer* (BES) using three different pre-trained transformer models: **BERT**, **GPT-2**, and **XLNet** on the **160K** sample, using three different *evaluation metrics*. We compared all the extracted summaries with their corresponding *period-restored* original *transcripts* (step 2 of Phase I) for computing their *BLEU* and *ROUGE scores* as *we* *didn't have 160K manual reference summaries to compare them with*. Hence, the scores were low as the compared texts were of unequal lengths. However, the situation was the same for all the compared summarizers and the objective was to determine the extent of overlap between the extracted summaries (in the four cases) and the original transcripts. The *BLEU and ROUGE scores*, for each type of transcript (chat/customer/agent), represent the *average* of the *BLEU* and *ROUGE* scores of all the summaries generated from the corresponding type of transcripts contained in the sample.

Table 1 shows that our own extractive summarizer generated chat, customer and agent summaries with higher *average BLEU* and *average ROUGE scores* than BES using the three pre-trained models: **BERT**, **GPT-2**, and **XLNet** in approximately $\frac{1}{5},\frac{1}{7},\frac{1}{7}$ of the time taken by BES in summarizing the transcripts separately in the three cases. This is because our method employed a faster, *embedding-based summarization* step (step 7) that reduced the search space for sentence selection. Thus, it establishes that our extractive summarizer is more *effective* and *efficient* than BES for chat transcripts. *So, it made sense to use our summaries for fine-tuning the abstractive summarizers in Phase II*. Table 1 further illustrates that extractive summarization with *channel separation* generated more coherent summaries than without separation (***ablation study***) and the customer summaries were the most effective. The *punctuation-restoration-accuracy* scores for chat, customer and agent summaries varied between $90-100\%$ in all cases. The proposed summarizer is highly parameterized and more flexible than *BERT Extractive Summarizer*.

Table 2 shows results from Phase II and compares the performances of the four fine-tuned abstractive summarizers, i.e., **T5**, **BART**, **Longformer2Roberta** and **DialogLED** on the hold-out set (**5K**). We compared all the customer and agent *abstracted* summaries with their corresponding *extracted summaries* from Phase I, for computing their *BLEU* and *ROUGE scores*. The scores were higher as texts were of comparable lengths. Table 2 shows that **BART** generated customer and agent abstractive summaries were *closest* to the extractive summaries with the highest *average BLEU* and *average ROUGE scores*, while taking the *least* fine-tuning time. Table 2 further confirms that **Longformer2Roberta** was more *effective* and *efficient* than **DialogLED** for our transcripts while **T5** was the least effective for chat transcript summarization. The **BART** models, fine-tuned on customer and agent transcripts, are being readied for production deployment.

ES	Chat BS	Chat RS	Cust. BS	Cust. RS	Agent BS	Agent RS	TST (secs)
IES	0.20	0.52	0.30	0.63	O.23	0.55	17,334 (~5 hours)
BES-1	0.13	0.44	0.27	0.59	0.16	0.47	85,867 (~24 hours)
BES-2	0.12	0.40	0.26	0.57	0.15	0.44	124,161 (~35 hours)
BES-3	0.12	0.41	0.26	0.58	0.15	0.45	118,199 (~33 hours)

Table 1: Metric scores for Extractive Summarizers.
[ES: Extractive Summarizer, BS: BLEU Score,
RS: ROUGE Score, TST: Total Summarization Time,
IES: Indigenous Extractive Summarizer,
BES-1: BES(BERT), BES-2: BES(GPT-2),
BES-3: BERT(XLNet)]

AS	Cust. BS	Cust. RS	Agent BS	Agent RS	AFS (secs)
T5	0.41	0.56	0.58	0.73	50,242 (~14 hours)
BART	0.62	0.72	0.67	0.83	19,214 (~5.34 hours)
Longformer 2Roberta	0.47	0.66	0.61	0.77	79,086 (~22 hours)
DialogLED	0.46	0.63	0.55	0.74	114,574 (~32 hours)

Table 2: Metric scores for Abstractive Summarizers.
[AS: Abstractive Summarizer, BS: BLEU Score,
RS: ROUGE Score, AFS: Average Fine-tuning Time]

Next, we present another ***ablation study*** on the impacts of *fine-tuning* on the language models. The four language-model (LM/LLM) based abstractive summarizers were also used to summarize (*zero-shot*) the chat transcripts in the hold-out set without *fine-tuning* any of the models, to measure the full impact of *fine tuning* for chat summarization in our context. Table 3 shows results from our ***ablation study*** and demonstrates that **DialogLED** performed the best on *BLEU* scores, while **Longformer2Roberta** performed the best for *ROUGE* scores amongst the four *untuned* abstractive summarizers for all transcript types. Untuned **BART** was the least effective. Furthermore, comparing metric scores in Table 2 and Table 3, we can conclude that on an average fine-tuning improved the performance of an abstractive summarizer on all chat transcripts by ~**8** times, on customer transcripts by ~**5** times, and on agent transcripts by ~**11** times. **BART** showed the most improvement in fine-tuning on our chat transcripts.

AS	Cust. BS	Cust. RS	Agent BS	Agent RS
T5	0.07	0.28	0.05	0.21
BART	0.03	0.21	0.01	0.16
Longformer2Roberta	0.09	0.38	0.08	0.33
DialogLED	0.17	0.36	0.14	0.29

Table 3: Metric scores in Ablation Study
(Impacts of Fine-tuning).
[AS: Abstractive Summarizer,
BS: BLEU Score, RS: ROUGE Score]

7.5 Limitations

There are two limitations associated with the proposed method, one related to its *evaluation procedure* and the other related to its *capability*. Limitation related to its automated evaluation originated from not having enough manually crafted reference summaries for the 160K chat transcripts under consideration. In the absence of a full set of reference summaries, we compared the extracted summaries with the *period-restored* and *longer* original transcripts (from step 2 of our extractive procedure) for computing their corresponding *BLEU* and *ROUGE scores*. So, the scores were slightly lower. However, this was done for the summaries from the proposed method as well as for the three pre-trained language model driven *Bert Extractive Summarizers* to ensure consistency and similarity in the comparisons. Likewise, in the absence of manually generated reference summaries, the abstractive summarizers were fine-tuned on the extractive summaries, and we automatically compared the abstractive summaries with the extractive summaries using commonly used metric scores. However, this was done only after verifying through both automatic and some manual evaluations that our extractive summaries were highly readable and usable. On the other hand, one limitation of its capability is that it *doesn't repair grammatical errors* (one of the challenges associated with the chat transcripts), only reduces their numbers with fewer sentences, some postprocessing and abstractive summarization through pre-trained language models. This also explains the rationale behind the use of the two *denoising* abstractive summarizers in Phase II for abstractive summarization.

Furthermore, it may be noted here that this research was started several years back, prior to the arrival of the latest generation of prompt-based, all-purpose, decoder-transformer models, e.g., GPT-3, ChatGPT (GPT-3.5/4), Llama, Gemini, etc., which can also be quite effective for *zero-shot* text summarization (Zhang et al., 2024). Consequently, for this version, we considered and tested slightly earlier generation of LLMs/LMs, i.e., non-instruction-based encoder-decoder models, which had been previously used in the literature and are still utilized widely for text (e.g., dialog) summarization (generation), specifically for more customized use cases.

8 Conclusion

In this paper, we have presented a hybrid summarization technique to address some of the challenges associated with chat transcript summarization, prevalent in our context. We have combined *channel separation*, *topic modeling*, *sentence selection*, *punctuation restoration*, in *extractive summarization*, with *transfer learning* based *supervised abstractive summarization*, to generate coherent and more readable chat transcript summaries for a better understanding of the customer complaints and the agent resolutions. The proposed summarizer is the *only* hybrid one that restores *full punctuation* to the summaries. Finally, we have established the efficacy of the hybrid strategy through extensive experimentations and performance comparisons. The hybrid method is very useful for large-scale deployment of chat transcript summarization, in the absence of *manually* crafted reference (annotated) summaries for fine-tuning the abstractive summarizers.

References

Sanghwan Bae, Taeuk Kim, Jihoon Kim, Sang-goo Lee. 2019. Summary Level Training of Sentence Rewriting for Abstractive Summarization. *arXiv preprint arXiv:1909.08752v3*.

Itz Beltazi, Matthew E. Peters, Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150v2*.

Pratik K. Biswas, Aleksandr Iakubovich. 2022. Extractive Summarization of Call transcripts. In *IEEE Access*, vol. 10, pages 119826-119840.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175v2*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of deep bi-directional transformers for language understanding. *arXiv preprint arXiv:1810.04805v2*.

Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, Dragomir Radev. 2021. ConvoSumm: Conversation Summarization Benchmark and Improved Abstractive Summarization with Argument Mining. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on NLP (Volume 1: Long Papers)*, pages 6866-6880.

Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, Ranit Aharonov. 2021. TWEETSUMM - A Dialog Summarization Dataset for Customer Service. In *Findings of the ACL: EMNLP 2021*, pages 245-260.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 19–25.

Karel Jezek and Josef Steinberger. 2008. Automatic Text Summarization. 2008. *FIIT STU Brarislava, Ustav Informatiky a softveroveho inzinierstva*. Vaclav Snasel (Ed.), pages 1- 12.

Karen Sparck Jones. 2007. Automatic summarizing: The state of the art. *Information Processing & Management*, vol. 43, no. 6, pages 1449–1481.

Atif Khan and Naomie Salim. 2014. A review on abstractive summarization Methods**.** *Journal of Theoretical and Applied Information Technology*, 2014, vol. 59, no. 1.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, & Comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL Workshop*, Barcelona, Spain, pages 74-81.

Hui Lin and Vincent Ng. 2009. Abstractive summarization: a survey of the state of the art. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pages 9815-9822.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li, 2017. Generative Adversarial Network for Abstractive Text Summarization. *arXiv preprint arXiv:1711.09357v1*.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318v2*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of EMNLP-IJCNLP*, pages 3730-3740.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development,* vol. 2, no. 2, pages 159–165.

Inderjeet Mani. 2001. Automatic Summarization. Natural language processing. John Benjamins Publishing Company.

Derek Miller. 2019. Leveraging BERT for text summarization on lectures. *arXiv preprint arXiv:1906.04165v1*.

Ramesh Nallapati, Bowen Zhou, C´ıcero Nogueira dos Santos, C¸ aglar Gulc'ehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany, pages 280– 290.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *In Mining Text Data*. Springer, pages 43–76.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th ACL*, Philadelphia, USA, pages 311-318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational linguistics,* vol. 28, no. 4, pages 399–408.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, Multilingual Information Extraction and Summarization*. Springer, pages 3–21.

Abigail See, Peter J. Liu, Christopher D. Manning. 2017. Get to the point: summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368v2*.

Ming-Hsiang Su, Chung-Hsien Wu, Hao-Tse Cheng. 2020. A two-stage Transformer-based Approach for Variable Length Abstractive Summarization. *IEEE/ACM TASLP*, vol. 28, pages 2061-2072.

Don Tuggener, Margot Mieskes, Jan Deriu, and Mark Cieliebak. 2021. Are We Summarizing the Right Way? A Survey of Dialogue Summarization Data Sets. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 107–118.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP Conference: Short Papers*, pages 297-300.

Klaus Zechner. 1997. A Literature Survey on Information Extraction and Text Summarization. Computational Linguistics Program, Carnegie Mellon University, April 14.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, vol. 12, pages 39–57.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. DialogLM: Pre-trained Model for Long Dialogue Understanding and Summarization. *arXiv preprint arXiv:2109.02492*.

Real-Time Speech-Driven Avatar Animation by Predicting Facial landmarks and Deformation Blendshapes

Juan Camilo Vásquez-Correa[1], **Santiago Moreno-Acevedo**[1], **Ander Gonzalez-Docasal**[1],
Aritz Lasarguren[2], **Jone López**[2], **Egoitz Rodriguez**[2] and **Aitor Álvarez**[1]
[1]Fundación Vicomtech, Basque Research and Technology Alliance (BRTA),
Donostia – San Sebastian, Spain
[2]Baleuko S L, Durango, Spain
`jcvasquez@vicomtech.org`

Abstract

The evolution of virtual spaces and live events demands sophisticated methods for avatar animation. While existing techniques offer diverse approaches, limitations persist in achieving real-time responsiveness and natural communication. This paper proposes a novel approach for real-time speech-driven avatar animation, covering the prediction of 2D and 3D facial landmarks, and deformation blendshapes from ARkit. Specific models were trained to generate both emotional and neutral animated faces, and using convolutional neural networks able to deal with low latency requirements. The quality of the generated animations was addressed both objectively and subjectively. Both evaluations suggest that our approach is accurate to generate high-fidelity and expressive animations. In addition, we create a client-server application that achieved real time performance, enabling frame rates and latencies suitable for live interactions, fostering a seamless and immersive experience.

1 Introduction

Modern animated movies and games rely on expressive facial animation to convey emotions and enhance storytelling. While vision-based technology plays a vital role in capturing real actors' performances and translating them onto animated characters, it often comes at a significant cost (Karras et al., 2017). Elaborated hardware setups are frequently required for computer vision systems, and re-shoots necessitate the actors' physical presence and consistent appearance. Conversely, speech-driven algorithms are a compelling alternative by significantly reducing costs. For instance, animating vast amounts of in-game dialogue becomes significantly cheaper through audio processing instead of costly video capture setups (Karras et al., 2017). Additionally, speech-driven systems can leverage natural animations even from Text-to-Speech models, opening up new possibilities for character creation.

When generating facial animations from speech, it is important not only to ensure lip-sync, but also to transfer the emotions of the speaker into the avatar to guarantee a more natural communication (Chen et al., 2023). Humans are experts in facial reading, making inconsistencies between speech and facial expression to be potentially distracting, unpleasant, and even confusing. This is evident in the McGurk effect, where mismatched visual and auditory speech can alter perceived words (Alsius et al., 2018). Therefore, high-fidelity speech animation becomes essential for conveying emotions, intentions, and creating truly immersive experiences.

Speech facial animation technologies fall into two broad categories based on complexity and expressiveness. Some engines leverage large-scale neural models for highly nuanced animation, as described by Yang et al. (2023); Zhao et al. (2024). However, these solutions often demand significant computational resources, limiting their suitability for resource-constrained projects. On the contrary, simpler libraries based on viseme recognition (Edwards et al., 2016) offer faster animation, but are often criticized for lacking emotional expressiveness and intent transfer (Taylor et al., 2017).

Despite advancements in speech-driven animation, achieving real-time performance and seamless integration with animation software remains a challenge. Current systems are based on facial landmark predictions (Taylor et al., 2017; Eskimez et al., 2019; Vidal and Busso, 2023) and 3D facial meshes (Chen et al., 2023; Thambiraja et al., 2023; Zhao et al., 2024), which are able to produce high fidelity and natural animations. However, they have limitations in computational efficiency and software compatibility. Additionally, approaches directly mapping speech to video animations (R. et al., 2023; Zhang et al., 2024) often prioritize expressiveness over real-time performance, hindering practical applications. Existing models rely on large-scale architectures like Recurrent Neural Networks (RNNs) (Pham et al., 2017; Eskimez et al., 2018; Y. et al., 2020; Zhou et al., 2020; Villanueva et al., 2022), Transformer models (Chen et al., 2023; Yang et al., 2023; Xing et al., 2023; Zhang et al., 2023), diffusion models (Thambiraja et al., 2023), and Generative Adversarial Networks (GANs) (K. and E., 2021; Zhang et al., 2021; Vougioukas et al., 2020). RNNs effectively model temporal dependencies in speech, transformers excel at long-range context analysis, and GANs can generate highly natural animations. Although current techniques offer high quality animations, they struggle to achieve the real-time responsiveness and natural communication cues necessary for truly immersive experiences.

The growing sophistication of virtual spaces and interactive live events demands new methods for avatar animation that go beyond high fidelity. This paper addresses current limitations by proposing a novel, real-time speech-driven avatar animation engine to bridge the gap between high-fidelity visuals and smooth interaction during interactive live events. We considered deep architectures to generate several animation representation types in real time, including 2D/3D facial landmarks, and ARkit deformation blendshapes[1]. Several studies have addressed the prediction of facial landmarks based on speech (Taylor et al., 2017; Zhou et al., 2020) as part of their pipelines. Studies relying on ARkit blendhapes have focused on performing audiovisual speech synthesis, using adaptations of Tacotron2 (Hussen Abdelaziz et al., 2021). However, such approaches limits both the emotional expressiveness that real actors can transmit to the generated faces. To the best of our knowledge, this is one of the first studies focused on predicting ARkit blendshapes directly from speech, and the first one aiming to generate them in real time, paving the way for expressive and interactive avatars during live animation events.

The performance of the proposed models is evaluated both objectively and subjectively in order to check not only the accuracy of the predicted landmarks and blendshapes, but also perceptual indicators about expressiveness, coherence, quality, and lip-sync. In particular, subjective tests are conducted by a group of 3D animation experts, increasing the novelty of the proposed approach with respect to related studies that have perform subjective tests only with naive users (Y. et al., 2020). Furthermore, we performed an extensive evaluation of the run-time capabilities that are essential for real-time animation production in live events. Unlike previous studies focusing only on limited audio samples and single frame prediction times (Tian et al., 2019; Lu et al., 2021), our work provides a more comprehensive assessment in production-ready environments typically found in live events.

2 Methods

2.1 Facial Animation Representations

We incorporated three animation representation types to address different application scenarios when animating avatars: (1) 2D facial landmarks, (2) 3D facial landmarks, and (3) deformation blendshapes. These animation types are intended to be transmitted in real time to animation engines like Unity[2], Blender[3], or Maya[4] to animate cartoon-type avatars that follow the facial expressions of an actor. Each representation is considered depending on the type and realism of the avatar to be animated.

Facial landmarks are key reference points on a face, used to track movement, expression, and individual facial structures on a coordinate system. We considered both 2D and 3D facial landmark representations that are automatically extracted from video frames. The 2D landmark points correspond to 68 x-y coordinates extracted using the DLib library (King, 2009), and which have been used in similar studies to map the general facial structure (R. et al., 2023; Eskimez et al., 2018) (see Figure 1a). The 3D landmark representation consists of 478 x-y-z coordinates extracted using the MediaPipe Facemesh model from Google (Grishchenko et al., 2020; Yan, 2022), and which is able to extract more fine-grained information from the facial structure and map it into more realistic 3D avatars (see Figure 1b).

Complementary to facial landmarks, blendshapes are pre-sculpted variations of an object e.g., the face, used to smoothly animate complex deformations of its geometry. Blendshapes are standard animation mechanisms widely used in professional animation engines. We considered a stan-

---

[1] `https://developer.apple.com/documentation/arkit/`

[2] `https://unity.com/`

[3] `https://www.blender.org/`

[4] `https://www.autodesk.es/products/maya`

dard set of 52 ARKit blendshapes[5] that allow to animate the eyebrows, mouth, jaw, and lips in different ways, and adapt the coefficients to a specific avatar (Figures 1c and 1d). The set of blendshapes was extracted using the MediaPipe Blendshape V2 model (Grishchenko et al., 2022).

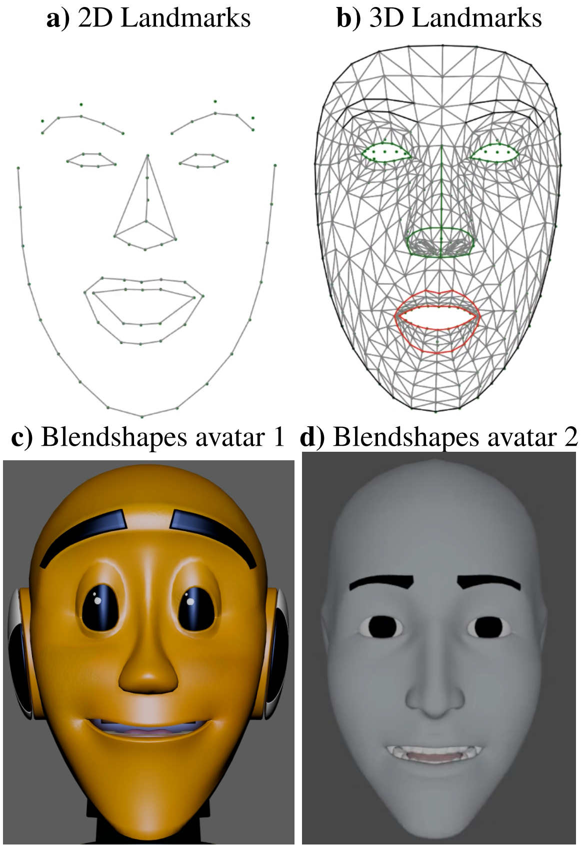

a) 2D Landmarks **b)** 3D Landmarks **c)** Blendshapes avatar 1 **d)** Blendshapes avatar 2

Figure 1: Facial representations for speech-driven avatar animation, covering facial landmarks and deformation blendshapes.

2.2 Deep Architectures

Several models for the three animation representations were trained using combinations of convolutional and recurrent networks. In particular, we considered the Long Short-Term Memory (LSTM) network proposed in (Eskimez et al., 2018) as a baseline, an adapted version of the 1D-CNN used in (Eskimez et al., 2019), and a CNN built on top of SincNet filters (Ravanelli and Bengio, 2018). These architectures were selected with the purpose of producing speech-driven animations in real time. As a consequence, more complex and bigger models like those based on diffusion (Zhao et al., 2024; Zhang et al., 2024; K. and H., 2023) or Transformers (Chen et al., 2023; Xing et al., 2023) were not considered.

The baseline model from (Eskimez et al., 2018) uses the first and second order temporal difference of log-Mel spectrograms as input of a four-layer LSTM network. This network was trained to generate 2D landmark points with a temporal resolution of 40 ms.

The second considered model is a 4-layer 1D-CNN (kernel size of 21 and number of channels=$\{64, 128, 256, 512\}$, respectively), adapted from (Eskimez et al., 2019), and which is trained to predict Point Distributed Models (PDMs) for 2D/3D landmarks, and the 52 ArKit blendshapes. PDMs reduce variability in landmark predictions due to face shape, scale, and orientation (Cootes et al., 1995). These PDMs are shape models that represent the high-dimensional landmark space with a set of coefficients obtained after PCA decomposition. The output of the last convolutional layer is finally processed by a linear layer to make the final predictions of the landmarks and blendshape coefficients. The CNN receives as input 280 ms of the raw speech waveform (7 frames of 40 ms) and predicts the PDM coefficients of the central frame, using the remaining frames as past and future context.

Finally, we propose the use of a SincNet model (Ravanelli and Bengio, 2018) trained also to predict the PDM coefficients for 2D/3D landmarks, and the blendshapes. Our model consists of a SincNet layer fed by 280 ms of the raw waveform and which generates speech tokens with a 40 ms resolution. The output of the SincNet layer is then processed by two convolutional layers and two linear layers to make the final prediction of the PDMs or the blendshapes.

For training all considered models, we employed the Smooth-L1 loss function and implemented a 5-fold speaker independent cross-validation strategy, using four folds for training and development, and the remaining one for independent testing. The models are trained sing Adam, with a batch size of 32 audio samples, a learning rate of 10^{-5} and dropout of 0.1. The dimension of the PDMs was set to 20 when predicting the landmarks, keeping 0.99 of the cumulative variance when computing PCA. The models were trained during 20 epochs.

Finally, to reduce high-frequency noise, particularly visible as tremors in the eyebrows and eyes, the predicted blendshapes undergo post-processing with a Savitzky-Golay filter (Schafer, 2011). This

---

[5] `https://arkit-face-blendshapes.com/`

filter smoothes the data while preserving underlying trends, resulting in more natural and visually appealing facial animations.

2.3 Real Time Processing

We developed a client-server application utilizing FFmpeg[6] and Websockets for real-time audio stream processing. The client transmits continuous audio streams of 1024 bytes (corresponding to 32 ms of audio sampled at 16 kHz and 16-bit resolution) to the server. The server continuously receives and buffers the stream, maintaining a processing buffer. Once the buffer reaches 280 ms (7 frames of 40 ms), the server predicts facial animations for the central frame and sends the results back to the client for visualization and integration with animation engines. After processing, the server releases the corresponding 40 ms audio segment from the buffer and waits for new frames to arrive. An overview of the processing setup is shown in Figure 2.

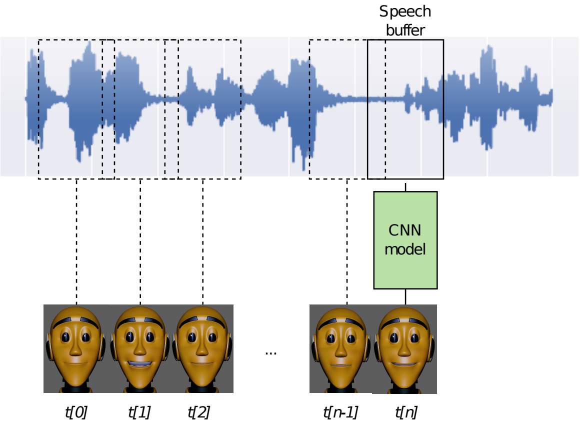


Figure 2: Overview of the proposed system for real time avatar animation. An animation is generated every 40 ms from a buffer size of 280 ms. This distribution guarantees a continuous stream of blendshapes and landmarks at 25 FPS, with a delay of 280 ms.

This configuration ensures a continuous stream of 2D/3D landmarks or blendshapes from the server to the client with an average rate of 25 frames per second (FPS) and a processing delay of 280 ms (reflecting the queue length used for context during prediction). Crucially, single frame processing time must be guaranteed to be less than 32 ms (duration of the received audio stream) to avoid queue buildup and maintain uninterrupted streaming. Sending larger audio chunks leads to faster queue filling and requires consecutive frame processing, potentially causing server response delays and packet loss due to queue overflow. Finally, with the aim to generate more natural animations, artificial blinks were introduced in the server predictions every 5 seconds (with a certain probability) by modifying the corresponding blendshape coefficients or the 2D/3D landmarks.

3 Data Description

The animation models were trained using the CREMA-D (Cao et al., 2014) and the Grid (Cooke et al., 2006) corpora. These datasets were selected with the aim to have individual models for emotional and neutral speech-driven facial animations (see Table 1). Both corpora have been used in similar studies, particularly in realistic talking face generation (Vougioukas et al., 2020; Kefalas et al., 2020). Labels for 2D/3D landmarks, and blendshape coefficients were extracted from videos using the methods described in Section 2.1.

	CREMA-D	**Grid**
Emotions	Six emotions	Neutral
# Utterances	7,442	34,000
Duration (hours)	6.2	28.3
# Sentences	12	1000
# Speakers	91	34
Camera	Panasonic AG-HPX170	Canon XM2
Video	Flash at 30 FPS 480x360	MPEG at 25 FPS 360x288

Table 1: Information of source corpora used to train the speech-driven facial animation models.

CREMA-D (Cao et al., 2014) is an emotional multimodal acted dataset, used traditionally for speech emotion recognition. Actors spoke a selection of 12 sentences in six emotions (Anger, Disgust, Fear, Happiness, Neutral, and Sadness) and three emotion levels (Low, Medium, High), in English. Models trained with this dataset will generate more expressive and emotional animations.

The Grid Corpus (Cooke et al., 2006) was designed for audiovisual speech recognition, in English language. The dataset includes high-quality audio and facial video recordings of 1,000 sentences spoken by 34 subjects (18 male, 16 female). The sentences spoken by each actor are composed of six words randomly chosen from a limited dictionary. Although this corpus has a restricted vocabulary, it was selected to facilitate the development of models capable of generating accurate animations with high lip-synchronization quality (Vougioukas et al., 2020) and to serve as a benchmark for measuring the potential performance limits of the trained models.

---

[6] `https://ffmpeg.org/`

4 Experiments and Results

4.1 Objective Evaluation

The 2D and 3D landmark prediction models were evaluated using different metrics from the literature (Zhou et al., 2020). In particular, we included the landmark distance (L-D), the landmark velocity difference (L-VD), the L-D for jaw-lips, and the difference in the open mouth area (OMA-D). We introduced additional metrics to specifically evaluate lip-sync quality such as the L-D and L-VD for mouth-specific landmarks.

Table 2 shows the quality evaluation of the models to predict 2D landmarks. Both 1D-CNN and SincNet networks achieved significantly lower errors than the baseline (for L-D related metrics), and especially for the landmarks defining the jaw-lips and the mouth, as demonstrated by one-way ANOVA with pairwise Tukey post hoc tests (p-value $\ll 0.005$). These results were observed for both neutral and emotional models trained with respective datasets (Grid and CREMA-D). For velocity related metrics that evaluate the temporal dynamics of the facial animations, the baseline models exhibited lower error rates. This is expected due to the nature of the LSTM-based model from the baseline, which is better to model temporal dependencies. However, note that the recurrent nature of such model make it not being able for real time processing, which is a key objective of this work. In addition, no differences were found between the 1D-CNN and SincNet predictions (p-value $= 0.652$). Finally, the models performed better at predicting facial landmarks for neutral faces than the emotional ones from the CREMA-D corpus (p-value $\ll 0.005$). A separate analysis showed that within the CREMA-D corpus, facial landmarks of low-arousal emotions (sadness and disgust) had lower prediction errors than high-arousal ones.

	CREMA-D			Grid		
Metric	**Baseline**	**1D-CNN**	**SincNet**	**Baseline**	**1D-CNN**	**SincNet**
L-D	1.28	1.06	1.20	0.81	0.52	0.52
L-VD	5.00	5.41	5.38	4.21	4.37	4.55
L-D jaw-lips	0.73	0.56	0.59	0.77	0.37	0.36
L-VD jaw-lips	4.81	5.55	5.58	4.53	5.11	5.09
OMA-D	0.51	0.31	0.72	0.31	0.16	0.17
L-D mouth	1.54	1.21	1.56	1.00	0.54	0.55
L-VD mouth	5.79	6.11	6.36	4.49	4.63	4.99

Table 2: Error metrics (%) for the prediction of 2D landmarks.

Table 3 presents the results predicting 3D facial landmarks. The baseline models was not considered here considering again that our ultimate goal is to perform real time predictions, which the baseline model is not able to achieve. The errors were higher than those reported for 2D landmarks, which is expected because the significantly larger number of points to predict (over 10 times more). However, the errors remained below 6 % for the entire set of landmarks, and below 4 % for the ones related to the mouth movement. In this case, the 1D-CNN model surpassed SincNet, with statistically significant lower errors (p-value $\ll 0.005$) for both neutral and emotional datasets.

	CREMA-D		Grid	
Metric	**1D-CNN**	**SincNet**	**1D-CNN**	**SincNet**
L-D	3.45	3.64	5.85	6.00
L-VD	9.92	13.1	6.32	7.47
L-D jaw-lips	1.54	1.83	5.97	5.92
L-VD jaw-lips	9.78	14.87	5.28	6.41
OMA-D	0.23	0.23	0.20	0.20
L-D mouth	3.51	3.69	3.75	3.79
L-VD mouth	10.74	14.73	8.03	9.88

Table 3: Error metrics (%) for the prediction of 3D landmarks.

Finally, the quality of the blendshape predictions is evaluated with the average Mean Absolute Error (MAE) of the 52 blendshapes, and subsets related to specific facial areas such as the mouth, cheeks, jaw, eyes, and eyebrows. The results are shown in Table 4. Similar to the 2D landmarks case, there were no significant differences between the predictions obtained with the 1D-CNN and the SincNet models. Moreover, the neutral blendshapes from the Grid corpus were more accurately predicted, similar also to the 2D-landmark scenario. Regarding the blendshape generation of specific parts of the face, the cheek and jaw areas were the most accurately modeled, while the eyes and eyebrows were the most challenging to predict.

	CREMA-D		Grid	
Face area	**1D-CNN**	**SincNet**	**1D-CNN**	**SincNet**
All	14.67	14.33	9.65	9.74
Mouth	13.78	13.55	8.14	8.12
Cheeks	1.21	0.61	1.15	0.43
Jaw	6.55	6.44	6.08	5.94
Eyes	19.26	19.10	15.01	15.44
Eyebrows	28.57	27.58	14.69	15.52

Table 4: MAE (%) for the prediction of Blendshape coefficients.

4.2 Subjective Evaluation

Even though the previous results can evaluate the deviation of the reconstructed landmarks and blendshapes from the ground truth values, they are not able to measure the subjective aspects that come naturally to human viewers. We considered the emotional 1D-CNN model trained to predict ARkit blendshapes, and generated 40 videos featuring two emotions (euphoria and fear) on two different avatars (Figures 1c and 1d). The videos were produced using 20 independent audio samples (10 per emotion) recorded by an actress. 23 participants, divided into two groups (12 naive users and 11 3D animation experts), rated the videos. Each video received scores from 1 to 5 (with higher scores indicating better quality) across four criteria: (1) expressiveness, (2) coherence between the emotions conveyed by speech and facial expressions, (3) quality, which refers to the global quality of the animation in terms of realism, fluency and precision, and (4) lip-synchronization to measure how well the lip movements of the speaker matches the corresponding audio. The results are shown in Figure 3.

Our results are consistent with those reported in similar studies when animation and lip-sync quality are subjectively rated (Y. et al., 2020). There were no significant differences in the scores between the two rater groups (Mann Whitney U test, p-value > 0.05), although we observed that 3D animation experts usually assign higher scores than naive users. This can be likely explained because they are more aware of the difficulties of creating high-quality animations. In terms of emotions, the scores assigned for fear were slightly higher. However, they did not differ significantly from the obtained for euphoria. Finally, we observed that the perceived quality depends on the selected avatar. Users rated the avatar 1 (yellow avatar in Figure 1c) significantly higher, mainly because it has less human-like features. Therefore, the importance of correctly producing visemes was less important, contrary to the avatar 2 (gray avatar in Figure 1d).

4.3 Real Time Performance

The application was tested in an experimental setting consisting of separate client and server machines connected via WiFi through a VPN. The hardware specifications for both machines are shown in Table 5.

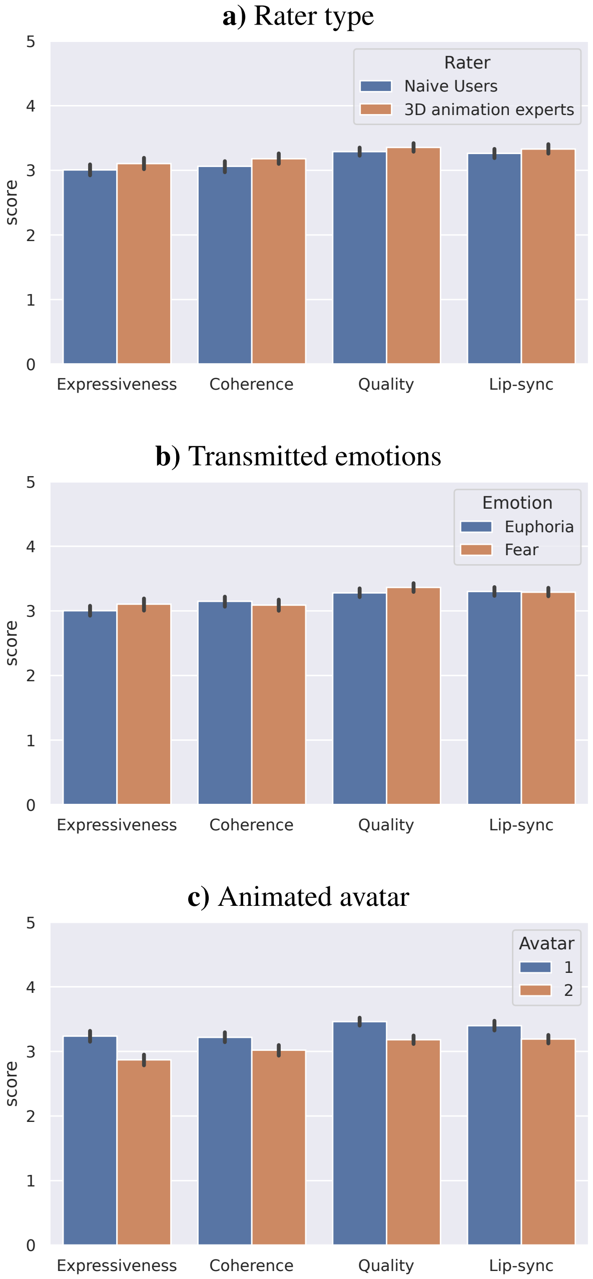


Figure 3: Subjective evaluations performed on the predicted ARkit blendshapes of 20 independent audio utterances recorded by an actress. The evaluations are discriminated in terms of the type of rater, the transmitted emotion, and the type of avatar.

	Client	**Server**
CPU	13th Gen Intel(R) Core i7-1355U 10 cores	Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz 16 cores (x2 Threads)
RAM	16 GB	128 GB
GPU	-	NVIDIA TITAN X (Pascal) 12GB

Table 5: Hardware specifications of the client and server machines for the real time evaluation

We evaluated the Real Time Factor (RTF) when predicting 2D landmarks from the CREMA-D corpus using the baseline, the 1D-CNN, and the SincNet models. The results are shown in Figure 4. Both the 1D-CNN and the SincNet models are suitable for real-time predictions as they achieved RTF $\ll 1$. Conversely, the recurrent nature of the LSTM model from the baseline resulted in an RTF > 1, making it unreliable for real-time predictions. Considering also that the 1D-CNN is the most accurate model for predicting landmarks and blendshapes, this model was used to test the reliability of a real application for performing avatar animations during continuous audio streams.

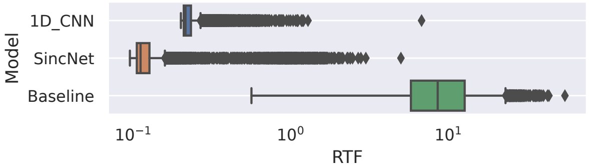


Figure 4: RTF when predicting 2D landmarks from the CREMA-D corpus using the baseline, the 1D-CNN, and the SincNet models.

To further evaluate the run-time performance, a one-hour speech stream was transmitted from client to server for real-time prediction of 2D/3D landmarks and blendshapes. Table 6 and Figure 5 summarize the performance in terms of several resources and quality metrics.

Run-time Metric	**2D Land.**	**3D Land.**	**Blendshapes**
RTF model prediction	0.07	0.14	0.07
Processed packages (%)	99.9	99.9	99.9
Maximum latency (ms)	21.2	22.6	19.7
Average FPS	24.9	25	24.9
Single frame processing time (ms)	2.77	5.84	2.66
Client RAM (MB)	182.1	183	181
Server RAM (MB)	480	477	489
Server GPU VRAM (MB)	2117	2139	2090
Queue time (ms)	285	260	283

Table 6: Runtime-performance of a continuous one-hour audio stream for real-time speech-driven facial avatar animation in terms of 2D/3D landmarks and ARkit blendshapes.

The system achieved real-time animation at 25 FPS with minimal latency (maximum of 22.6 ms) and no packet loss for all three scenarios. Individual frame processing consistently met the 32 ms requirement, ensuring uninterrupted streaming. Differences in processing time and RTF between 3D landmarks and the other animation modes arose from transforming predicted PDM coefficients into 3D landmarks (478 × 3 coordinates) and transmitting them back. While 3D landmarks required more computational and network resources than 2D landmarks and blendshapes, they did not hinder continuous transmission. Throughout the process, memory consumption remained low and stable across client, server RAM, and GPU memory. Notably, only 1/6th of GPU capacity was utilized, indicating potential cost reduction in future deployments.

The results obtained offer a more comprehensive overview of the requirements and run-time performance of a real application. Related studies that reported run-time performance have focused solely on generating predictions for a limited number of pre-existing audio samples, basing their conclusions exclusively on the time the model takes to predict a single animation frame (Tian et al., 2019; Lu et al., 2021). These studies did not consider critical factors during live events, such as connectivity issues, audio queuing, and memory overflow, which can occur during extended live transmissions.

5 Conclusion

We introduced a novel approach to produce facial animations in real time, specifically designed for interactive live events and virtual spaces. Different configurations of facial representations were considered, including 2D and 3D landmarks, and ARkit blendshapes, the latter one being a standard in professional animation engines. The modeling and prediction of the facial representations was performed using different configurations of CNNs due to the low latency requirements of the addressed application. The quality of the considered methods was evaluated both objectively using metrics from the state-of-the-art, and subjectively, where naive and expert raters estimated the quality of the generated animations. Finally, the best performing models were used to create a client-server application able to produce facial animations in real time.

The results indicated that both the 1D-CNN and the SincNet models were accurate enough to predict the three types of considered facial animations. The results also confirmed that it is more challenging to generate emotional facial animations than neutral ones. Additionally, the models demonstrated greater accuracy in predicting landmarks and blendshapes associated with mouth and jaw movements, compared to other facial regions like the eyebrows. Finally, the conducted runtime evaluations offer a broader understanding of real-time

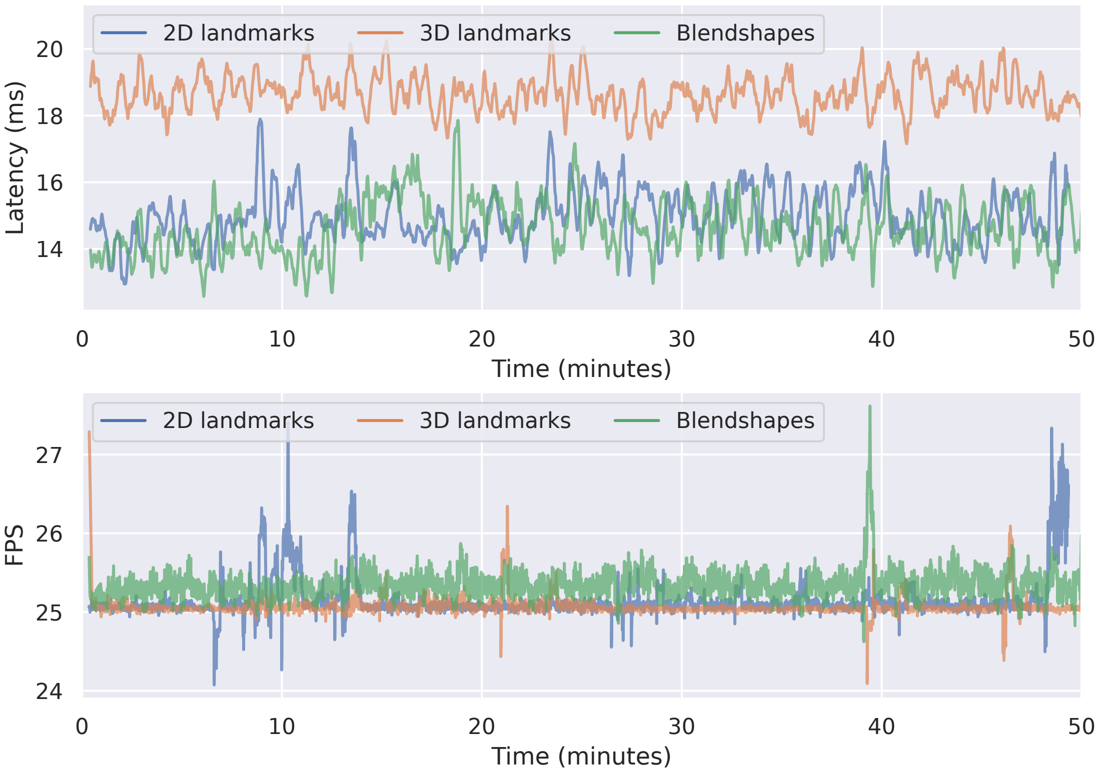


Figure 5: Latency and FPS of a continuous one hour audio streaming for the three different animation representation types.

application requirements and performance. Our real-time application showed that it is possible to generate facial landmarks and blendshapes in real-time at a constant rate of 25 FPS with a relative low latency and delay, and with low requirements of memory and GPU computation. Future work will be related to improve the quality of the generated animations in order to make them more natural and expressive. In this sense, novel architectures that also consider emotional classification can be proposed and evaluated. Exploring the integration of emotional intelligence into the system could be a promising direction for enhancing the expressiveness of the avatar animations.

Limitations

Despite the advancements and promising results from this paper, there are inherent limitations that should be considered: The first one is related to data availability. Although efforts were made to curate diverse datasets, the availability of comprehensive corpora covering a wide range of emotional expressions, linguistic diversity, and demographic variability might have been limited. This could potentially introduce issues in the generalization of the model to broader scenarios. Capturing the full spectrum of human emotions with high fidelity remains a challenge. Therefore, the current models may oversimplify the representation of certain emotional cues, leading to potential discrepancies between the intended and perceived expressions.

The second limitation relies on the latency and performance trade-offs when dealing with real-time applications. Achieving real-time responsiveness often requires optimizing for low latency, which may come at the expense of animation quality or computational resources. The study may have made certain compromises in this regard, and further optimizations could be explored to enhance the overall user experience.

Finally, while subjective evaluations provide valuable insights into the perceived quality of animations, they are inherently subjective and susceptible to biases. Factors such as individual preferences, cultural background, or expertise in animation could influence the raters' judgments. Employing diverse and representative rater groups, along with structured evaluation methodologies, can help mitigate bias to some extent but may not entirely eliminate it.

Ethics Statement

This study was conducted in accordance with the ACL Ethics Policy, ensuring that all research practices adhered to ethical standards in the development and evaluation of real-time speech-driven avatar animation.

In this study, all data used for training and evaluation were sourced from publicly available datasets, ensuring compliance with relevant data protection regulations. No personally identifiable information was used, and all data were anonymized to protect the privacy of individuals. Additionally, any data involving human participants was used in accordance with informed consent protocols.

While the technology developed in this study has positive applications, there is a potential for misuse, such as in the creation of deepfakes or unauthorized use of avatars. We emphasize the importance of deploying this technology responsibly, with safeguards to prevent misuse.

Finally, regarding transparency and accountability, we have provided detailed descriptions of our methodologies and evaluation metrics to ensure replicability and accountability. We encourage the research community to engage with and scrutinize our work to foster improvements and address any ethical concerns.

Acknowledgements

This study has received funding from the Basque Government Hazitek 2023 programme under grant agreement No ZL-2023/00497, Project ANIMEX.

References

A. Alsius, M. Paré, and K. G. Munhall. 2018. Forty years after hearing lips and seeing voices: The mcgurk effect revisited. *Multisensory Research*, 31(1-2):111–144.

H. Cao et al. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390.

Y. Chen, J. Zhao, and W. Zhang. 2023. Expressive speech-driven facial animation with controllable emotions. *arXiv preprint arXiv:2301.02008*.

M. Cooke et al. 2006. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424.

T. F Cootes et al. 1995. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.

P. Edwards et al. 2016. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG)*, 35(4):1–11.

S. E. Eskimez et al. 2018. Generating talking face landmarks from speech. In *Latent Variable Analysis and Signal Separation*, pages 372–381. Springer.

S. E. Eskimez et al. 2019. Noise-resilient training method for face landmark generation from speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:27–38.

I. Grishchenko et al. 2020. Attention mesh: High-fidelity face mesh prediction in real-time. *arXiv preprint arXiv:2006.10962*.

I. Grishchenko et al. 2022. Mediapipe blendshape v2 model card.

A. Hussen Abdelaziz et al. 2021. Audiovisual speech synthesis using tacotron2. In *Proc. International Conference on Multimodal Interaction*, pages 503–511.

Ege K. and Engin E. 2021. Investigating Contributions of Speech and Facial Landmarks for Talking Head Generation. In *Proc. Interspeech 2021*, pages 1624–1628.

Shuhei K. and Taiichi H. 2023. Speech-to-Face Conversion Using Denoising Diffusion Probabilistic Models. In *Proc. Interspeech*, pages 2188–2192.

T. Karras et al. 2017. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):1–12.

T. Kefalas et al. 2020. Speech-driven facial animation using polynomial fusion of features. In *Proc. ICASSP*, pages 3487–3491. IEEE.

D. E. King. 2009. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758.

Y. Lu et al. 2021. Live speech portraits: real-time photorealistic talking-head animation. *ACM Transactions on Graphics (TOG)*, 40(6):1–17.

H. X. Pham et al. 2017. Speech-driven 3d facial animation with implicit emotional awareness: A deep learning approach. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 80–88.

Xin R. et al. 2023. Emotion-Aware Audio-Driven Face Animation via Contrastive Feature Disentanglement. In *Proc. INTERSPEECH 2023*, pages 2728–2732.

M. Ravanelli and Y. Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *Proc. SLT*, pages 1021–1028. IEEE.

R. W. Schafer. 2011. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, 28(4):111–117.

S. Taylor et al. 2017. A deep learning approach for generalized speech animation. *ACM Transactions On Graphics (TOG)*, 36(4):1–11.

B. Thambiraja et al. 2023. 3diface: Diffusion-based speech-driven 3d facial animation and editing. *arXiv preprint arXiv:2312.00870*.

G. Tian et al. 2019. Audio2face: Generating speech/face animation from single audio with attention-based bidirectional lstm networks. In *Proc. international conference on Multimedia & Expo Workshops (ICMEW)*, pages 366–371. IEEE.

A. Vidal and C. Busso. 2023. Multimodal attention for lip synthesis using conditional generative adversarial networks. *Speech Communication*, 153:102959.

A. Villanueva et al. 2022. Voice2face: Audio-driven facial and tongue rig animations with cvaes. In *Computer Graphics Forum*, volume 41, pages 255–265. Wiley Online Library.

K. Vougioukas et al. 2020. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, 128(5):1398–1413.

J. Xing et al. 2023. Codetalker: Speech-driven 3d facial animation with discrete motion prior. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12780–12790.

Ravindra Y. et al. 2020. Stochastic Talking Face Generation Using Latent Distribution Matching. In *Proc. Interspeech 2020*, pages 1311–1315.

G. Yan. 2022. Mediapipe facemesh model card.

K. D. Yang et al. 2023. Probabilistic speech-driven 3d facial motion synthesis: New benchmarks, methods, and applications. *arXiv preprint arXiv:2311.18168*.

B. Zhang et al. 2024. Emotalker: Emotionally editable talking face generation via diffusion model. *arXiv preprint arXiv:2401.08049*.

C. Zhang et al. 2021. 3d talking face with personalized pose dynamics. *IEEE Transactions on Visualization and Computer Graphics*.

C. Zhang et al. 2023. Dream-talk: Diffusion-based realistic emotional audio-driven method for single image talking face generation. *arXiv preprint arXiv:2312.13578*.

Q. Zhao et al. 2024. Media2face: Co-speech facial animation generation with multi-modality guidance. *arXiv preprint arXiv:2401.15687*.

Y. Zhou et al. 2020. MakeItTalk: speaker-aware talking-head animation. *ACM Transactions On Graphics (TOG)*, 39(6):1–15.

Speech Emotion Recognition for Call Centers using Self-supervised Models: A Complete Pipeline for Industrial Applications

Juan M. Martín-Doñas[1] and **Asier López**[2] and **Mikel de Velasco**[2]
and **Juan C. Vásquez-Correa**[1] and **Aitor Álvarez**[1] and **María I. Torres**[2]
and **Paz Delgado**[3] and **Ane Lazpiur**[3] and **Blanca Romero**[3] and **Irati Alkorta**[4]
[1]Fundación Vicomtech, Basque Research and Technology Alliance (BRTA)
[2]University of the Basque Country UPV/EHU
[3]NaturalSpeech, [4]Gureak Marketing
`jmmartin@vicomtech.org`

Abstract

This paper presents a practical methodology to build adapted speech emotion recognition systems in call center scenarios for practical industrial applications. We focus on two specific use cases involving Spanish call centers with different characteristics in order to detect emotional states and improve their protocols. We address all stages of the development process, covering data acquisition, annotation, data harmonization, and model training and evaluation. We rely on cutting-edge self-supervised speech models for classification. This process has been designed to cover an industrial application's needs: data anonymity, reduced costs, and production-level performance. We compare the evaluated methods with well-established research benchmarks to validate our methodology. In addition, a subjective evaluation is performed to analyze their potential use in practical cases. The considered approaches show potential transferable results for these companies in their target call center scenarios.

1 Introduction

Call centers (CC) are increasingly leveraging speech analytics software to automate tasks and extract valuable insights from customer interactions (Hildebrand et al., 2020). This cutting-edge technology analyzes call recordings, enabling companies to enhance their operations. A pivotal aspect of CC conversations involves the speech emotion recognition (SER) of both clients and agents. This paralinguistic information can be used to efficiently transfer a voice call to a physical agent for further queries and discussions, to detect lies or even to identify emotional changes and states (Hema and Marquez, 2023). For instance, promptly identifying frustration can enable agents to employ de-escalation techniques or expedite issue resolution. Moreover, the data obtained can unveil broader trends in customer communication, empowering companies to refine their communication strategies. Therefore, developing reliable SER models holds immense value for the CC market, facilitating deeper customer insights, improved communication strategies, reduced customer frustration, and ultimately, a more positive customer experience (Irastorza and Torres, 2016, 2019).

Recent research on SER has focused on end-to-end deep learning systems, where self-supervised models have shown state-of-the-art (SOTA) performance in common benchmarks (Mohamed et al., 2022). These advancements have primarily been showcased in systems evaluated using acted or elicited databases (Busso et al., 2008). However, the efficacy tends to diminish considerably when these systems are deployed in real-world scenarios characterized by natural speech patterns (Zhu-Zhou et al., 2022). Notably, few works have explored the application of recent advances in SER within real call center environments. The work presented in (Bojanić et al., 2020) is an example of the SER technology application in a prioritizing urgency call system, which was evaluated on an acted Serbian corpus. Deschamps-Berger et al. (2021) evaluated convolutional-recurrent architectures for SER on the French CEMO corpus for medical emergency calls. For the customer service scenario, Pérez-Toro et al. (2021) proposed the classification of emotional states mapped on the arousal-valence dimensions to detect customer satisfaction using acoustic and linguistic models. On the contrary, Parra-Gallego and Orozco-Arroyave (2022) explored the evaluation of prosody and speaker embeddings to detect emotions and customer satisfaction in voicemails. Moreover, Feng and Devillers (2023) studied the continuous SER problem and the use of contextual information during the conversations. Płaza et al. (2022) addressed the database design and the development of solutions for SER classification, focusing on feature extraction methods to model both speech and text data using small classifiers. More recent studies (Deschamps-Berger et al., 2023; Macary et al., 2023) have proposed using SOTA self-supervised acoustic and language models for SER in CC scenarios, evaluating the performance in French SER corpus for research.

In summary, current technology exhibits considerable potential for enhancing call centers. Nevertheless, its successful implementation in real-world settings needs the resolution of substantial challenges. The establishment of a comprehensive database tailored to specific applications, the creation of precise classification systems or dealing with aspects of conversational speech remain great challenges that demand more research.

This study presents a novel methodology to either build and transfer an adapted SER solution to a real CC application. To this end, we collaborated with two companies providing call center services and designed a customized system for each to meet their specific needs within their respective domains. During the process, we covered the different stages of a suitable system design, starting from the domain data acquisition and annotation, including data pre-processing, manual labeling, and revision of the final corpus. Afterwards, we analysed and tested different feature extraction techniques for classification, from some more traditional to the most recent based on acoustic foundation models, and evaluated several downstream models focused on machine learning and deep learning techniques. In order to deploy practical systems for real scenarios when certain classes are under-represented, we also evaluate the application of binary detection systems that discriminates between neutral and emotional states, showing competitive performance and higher accuracy detection as it is requested in a practical environment.

Our work is complemented by evaluating the proposed solutions on well-established research databases to show that our systems follows the state of the art in the field. Moreover, a subjective comparison between manual and automatic emotional analysis is performed to assess the practical usability in potential real uses cases. Despite the high challenges of the task, our results demonstrate the successful development of transferable SER solutions, addressing the specific needs of the companies involved and holding significant promise for real-world applications.

The rest of the paper is organized as follows. Section 2 analyzes the process for acquiring, processing, and annotating the speech-emotional corpus created in this project. Then, we describe the experimental framework and evaluation results in Section 3. Finally, Section 4 summarizes the conclusions and possible research lines for future work.

2 Building a SER corpus for Call Centers

2.1 Tasks

In this project, we have collaborated with two Spanish companies providing CC solutions to build SER systems adapted to their unique needs and help them improve their interaction protocols and internal quality processes.

Since each company is dedicated to a different market, we tackled the SER problem from two different CC contexts. The first use-case (*CC-Support*) focuses on customer support, where customers often encounter issues like electronic signatures or login problems. The second one (*CC-Debt*) contains phone calls about debt collection, where stronger and more negative emotions usually arise. In this context, customers express frustration and anger more often than in CC-Support calls.

2.2 Data collection and annotation

The construction of the SER systems started with each call center providing approximately 60 hours of valuable speech in-domain data. Due to the calls' nature, containing personal and confidential information, the data could not be labeled in conventional platforms like Amazon Mechanical Turk[1]. Instead, the data were labeled by annotators trained for the task. Consequently, based on the expertise of the annotators and the resources available, we decided to use two annotators per segment instead the conventional three or more (Busso et al., 2008; Parada-Cabaleiro et al., 2018; Vázquez et al., 2019; Fan et al., 2021; Paccotacya-Yanque et al., 2022). This way, we prioritized data quantity over in-depth annotation. This approach would ultimately benefit the performance of our classifiers. The final amount of labeled data was determined by the budget allocated by each company for this task.

With the aim of speeding up the annotation process, the collected raw data were first preprocessed with a speaker diarization module (Landini et al., 2022) to separate the clients' and agents' speech. We discarded the segments with speaker overlap as well as segments shorter than 2.5 seconds, which often do not contain enough information to infer the emotional state of the speaker (Tóth et al., 2008).

---

[1] https://www.mturk.com/

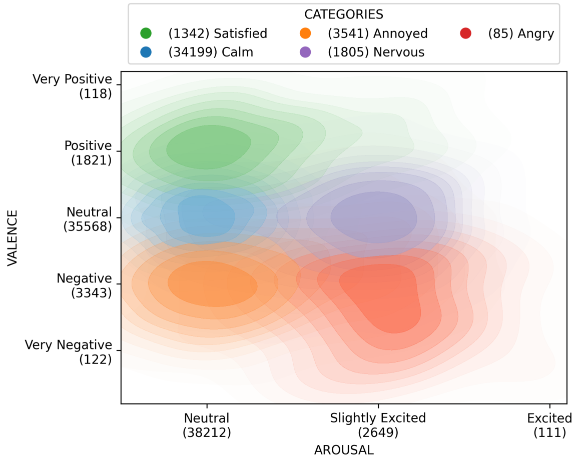


Figure 1: 2D Density Plot of the CC-Support data that shows the most prominent categories in the VA space.

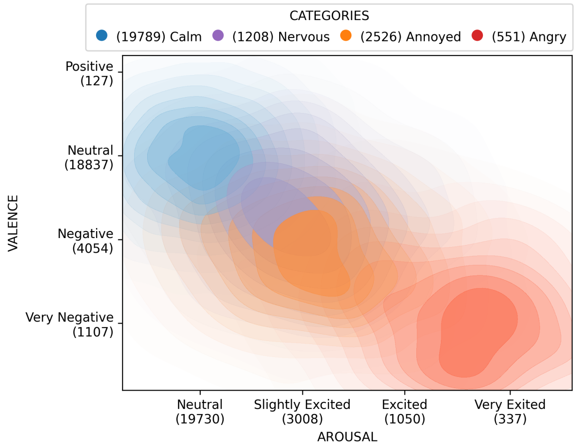


Figure 2: 2D Density Plot of the CC-Debt data that shows the most prominent categories in the VA space.

Finally, we split long turns into several less-than-20-seconds-long segments. This whole process led to an average of 7.3 and 9.1 seconds length segments for CC-Support and CC-Debt, respectively.

Once the preprocessed material from the raw data was prepared, several meetings with the CC annotators were held to establish the labelling criteria. The manual annotation was carried out through an application based on Praat (Boersma, 2001). Regarding the emotion representations, we defined them through the categorical (Ekman et al., 1999; Plutchik, 1980) and the Valence-Arousal dimensional model (Russell, 1980), which was discretized as in (de Velasco et al., 2022). We defined these choices to label each segment:

- Categories: Calm, Nervous, Angry, Annoyed, Surprised, Satisfied.
- Valence: Very Negative, Negative, Neutral, Positive, Very Positive.
- Arousal: Neutral, Slightly Excited, Excited, Very Excited.

2.3 Data preparation and analysis

Once the manual annotation was performed, we computed some data statistics to establish the ground truth labels, defining the following criteria in order to take out the most of our data:

1. If the amount of instances for a particular class is too low, discard them. For example, we removed the “Surprised" label in both use cases due to there were less than 30 labeled segments (combined annotations). The “Satisfied" class was also excluded from the CC-Debt use case for the same reason.
2. Merge two labels if they show a high correlation. For instance, we merged “Annoyed" with “Angry" in both cases.
3. Finally, speech samples where the two annotations differed was further analyzed in a process that involved the experts of the CCs[2]. The vast majority of disagreements were Neutral vs. Emotional labels. In these cases, we selected the Emotional label as the ground truth. Alternatives like discarding these samples, led to overall worse results.

In order to analyze the relation between the categorical emotions, we computed density plots in the arousal-valence plane, as shown in Figures 1 and 2. The numbers in brackets indicate the amount of segments per label (before merging). These maps reveal even more information, such as Annoyed and Angry were very related, in both cases. This phenomena was further noticed in preliminary classification experiments, where there was a noticeably high confusion between both emotions. Therefore, we merged them for our experiments. Similarly, we combined the Positive/Very Positive valences into a single label, the Slightly Excited/Excited arousal

---

[2] This was only needed for a small proportion of the dataset, because the overall agreement accuracy between the two annotations was 0.82 for CC-Support and 0.90 for CC-Debt, with average Cohen’s kappa coefficients of 0.16 and 0.67, respectively.

Table 1: Amount of data after annotation and post-processing for the CC-Support dataset. Note that some labels have been merged, and others discarded. Complete (Full) and binary (Bin) settings are indicated.

Dim.	Label	Samples	Time (h)	
			Full	Bin
Category	Calm	14817	30.2	30.2
	Annoyed	2723	6.2	11.3
	Nervous	1717	3.2	
	Satisfied	1241	1.9	
Valence	Neutral	15754	31.9	34.7
	Positive	1718	2.8	
	Negative	3026	6.8	6.8
Arousal	Neutral	18025	36.7	
	Excited	2473	4.8	

labels in CC-Support, and the Positive/Neutral valence labels and Excited/Very Excited arousal labels in the CC-Debt use case.

2.4 Final datasets

After the described post-process, we ended up with the amount of data shown in Tables 1 and 2 for CC-Support and CC-Debt, respectively. The information is given per dimension and class. The total amount of valid labeled hours for the CC-Support and CC-Debt use cases reached 41.5 and 30.5 hours respectively, where Calm and Neutral predominated over the emotional classes.

Finally, since one of the main objectives of the SER technology is to detect conflict points during the calls, we also created a more straightforward dataset, where each dimension (categorical, valence, and arousal) is composed of only two classes: Neutral and Emotional. To this end, we kept the majority class as Neutral, whereas the minority classes were merged into the Emotional class. We also indicated in Tables 1 and 2 the resulting amount of hours for this binary setting. This strategy is intended to improve the performance of the models by not only reducing the number of classes but also the imbalance of the data. This decision was jointly taken with the CC experts, as it was determined that different binary classifiers per dimension would provide enough information to assess whether a call should be carefully analysed.

Table 2: Amount of data after annotation and post-processing for the CC-Debt dataset. Note that some labels have been merged, and others discarded. Complete (Full) and binary (Bin) settings are indicated.

Dim.	Label	Samples	Time (h)	
			Full	Bin
Category	Calm	9476	23.0	23.0
	Annoyed	1717	5.1	7.5
	Nervous	873	2.4	
Valence	Neutral	9039	21.8	21.8
	Negative	2379	6.8	8.7
	Very Neg.	651	1.9	
Arousal	Neutral	10145	22.4	22.4
	Slig. Exc.	2030	5.8	8.1
	Excited	749	2.3	

3 Experimental results

3.1 Experimental framework

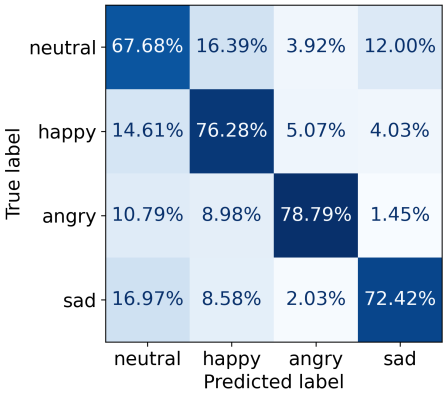
The constructed SER systems were evaluated on the real CC databases, in addition to the IEMOCAP corpus (Busso et al., 2008), in order to compare their performance in a well-established research database in the community. IEMOCAP consists of five dyadic sessions with two actors (male and female), summing up speech recordings that last nearly 12 hours. Following previous works (Pepino et al., 2021), we only evaluated categorical classification considering four different emotional classes: Anger, Happiness, Neutral, and Sadness.

All the experiments and evaluations were performed with a 5-fold cross-validation technique. For our in-domain databases, we split the recordings into five separate sessions, ensuring balanced (stratified) labels and that the audio samples from the same conversation were not distributed in different folds. Regarding the IEMOCAP corpus, each fold corresponded to a different recording session.

Different kinds of input features were evaluated during our analysis. Traditional features in the SER research community were employed, including eGeMAPS (Eyben et al., 2015), Compare 2016 (Eyben et al., 2015), as well as prosodic features (Parra-Gallego and Orozco-Arroyave, 2022). We also considered SOTA deep features for the SER task. First, we evaluated x-vector embeddings from a ResNet trained for speaker verification (Landini et al., 2022) due to the capability of these models to summarize various paralinguistic factors.

Table 3: F1 results (and CI) for the evaluation on the IEMOCAP dataset. CI of results in **bold** overlap with that of the best resulting model (**<u>underlined</u>**).

Feature	Classifier	IEMOCAP
eGeMAPS	SVM	56.11 ± 3.31
ComPare16		59.35 ± 2.67
Prosody		46.51 ± 2.04
x-vector		59.39 ± 2.83
W2V2		**71.22** ± **2.48**
WavLM		**72.75** ± **2.83**
W2V2	DNN-SP	**71.52** ± **2.54**
	DNN-AttCP	68.07 ± 2.52
WavLM	DNN-SP	**71.98** ± **2.58**
	DNN-AttCP	**<u>73.80</u>** ± **2.35**



True label \ Predicted label	neutral	happy	angry	sad
neutral	67.68%	16.39%	3.92%	12.00%
happy	14.61%	76.28%	5.07%	4.03%
angry	10.79%	8.98%	78.79%	1.45%
sad	16.97%	8.58%	2.03%	72.42%

Figure 3: Confusion matrix for the IEMOCAP dataset using a WavLM feature extractor with DNN-AttCP downstream classifier.

Following the current trends, embeddings from self-supervised models such as Wav2Vec2 XLS-R (W2V2) (Babu et al., 2022) and WavLM (Chen et al., 2022) were also analyzed due to their SOTA performance. Finally, the usefulness of content information was also analyzed using linguistic features. To this end, the audio was first transcribed using a medium Whisper model (Radford et al., 2023) fine-tuned with 500 hours of Spanish telephonic speech. The obtained transcriptions were fed to a Spanish BERT model called BETO (Cañete et al., 2020) to compute contextual representations.

Previous features were used to train and test two different machine learning classifiers. The first model consists of support vector machines (SVM) using one-vs-rest classification. The SVM was trained using the radial basis kernel and a balanced class weighting. Moreover, the features were standard normalized using the training set statistics. An average pooling was done for the deep learning models that output temporal sequences to compute the utterance vector representation. Moreover, for the speech self-supervised models, we also considered the hidden layer representations, which are known to contain more discriminative paralinguistic information (wen Yang et al., 2021).

To complete our analysis, DNN downstream classifiers were trained on top of the speech self-supervised models. We followed the approach presented in (Stafylakis et al., 2023; Kakouros et al., 2023), based on pre-trained self-supervised models with a weighted sum of the hidden representations before feeding the fine-tuned downstream network. Two different classifiers were considered, both of them based on embedding computation and softmax classification. The first one performs a linear transformation for dimensionality reduction followed by a simple mean-std statistical pooling (SP), and it is trained using cross-entropy (CE) loss. On the other hand, the second classifier also considers channel dropout and an alternative attentive correlation pooling (AttCP). To compute the attention weights, multiple heads are employed, and the similarities are aggregated prior the corresponding softmax layer via LogSumExp function. Finally, the classifier is trained using the CE loss with label smoothing.

During training, an 80-20 train-development partition was considered for model validation. The ADAM optimizer was used with a learning rate of $3 \cdot 10^{-4}$. Finally, to overcome the imbalanced dataset issue, a down-sampling strategy was followed to reduce samples at each epoch and keep a balanced distribution.

3.2 Results and analysis

We evaluated the different approaches in the IEMOCAP and the CC corpora. For the former, a category-level classification system was built. Similarly, we built one classifier per dimension (i.e., categorical, arousal, and valence) with the CC corpora. Besides, results are reported for both complete and binary label settings. The approaches were evaluated in terms of the macro-averaged F1-score, which accounts for imbalanced datasets. To consider the statistical significance when comparing systems, we performed bootstrapping (Keller et al., 2005; Ferrer and Riera) on the pooled test results to obtain 95% confidence intervals (CI).

Table 4: F1 scores for the evaluation on the CC in-domain data considering the different dimensions. Both the complete and binary version scores are included (separated by /). CI of results in **bold** overlap with that of the best resulting model (**underlined**).

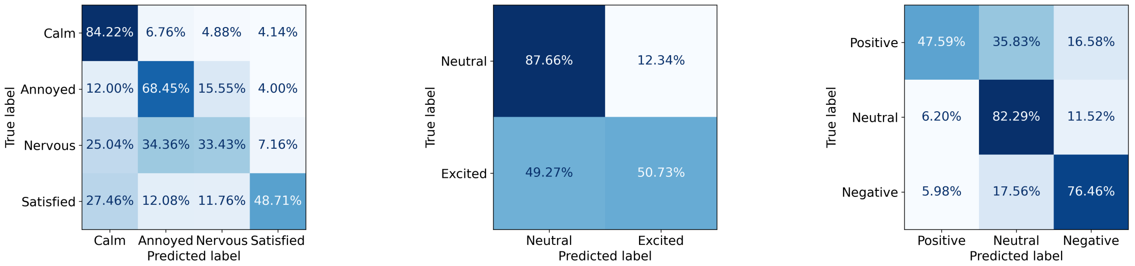
Feature	Classifier	CC-Debt			CC-Support		
		Category	Arousal	Valence	Category	Arousal	Valence
-	Random	26.25 / 45.60	26.40 / 46.05	26.70 / 46.60	19.15 / 47.35	41.60	26.75 / 42.90
	Majority	29.33 / 43.99	29.00 / 43.49	28.55 / 42.82	20.98 / 41.96	46.79	28.97 / 46.01
eGeMAPS	SVM	56.07 / 76.66	54.77 / 74.73	58.57 / 77.58	43.68 / 75.21	60.95	50.96 / 67.92
ComPare16		58.53 / 79.15	**56.18 / 77.28**	59.71 / 79.24	46.43 / 76.59	62.23	55.62 / 71.40
Prosody		49.82 / 72.22	49.54 / 70.90	51.07 / 72.10	37.58 / 68.86	57.71	46.04 / 63.89
x-vector		57.68 / 78.51	55.21 / 75.42	57.79 / 77.12	48.20 / 79.31	**64.41**	55.82 / 70.93
BETO		54.68 / 74.49	50.14 / 72.29	53.47 / 76.52	51.74 / 74.83	61.88	60.94 / **74.53**
W2V2		**64.35 / 82.63**	**62.05** / **80.32**	**66.18** / **82.41**	**56.04 / 81.78**	**66.14**	**64.43** / **77.08**
WavLM		**63.69 / 81.65**	**60.67 / 79.83**	**64.83 / 81.30**	**55.00 / 81.67**	**66.25**	**62.87 / 76.07**
W2V2	DNN-SP	**63.94 / 82.71**	**61.11 / 80.08**	**64.63 / 81.57**	**56.62 / 81.04**	**66.44**	**64.35 / 74.94**
	DNN-AttCP	**61.46 / 81.09**	**60.11 / 78.00**	**61.48 / 80.33**	**54.39** / 79.57	**64.60**	**62.47** / 73.95
WavLM	DNN-SP	**64.53 / 82.27**	**60.53 / 79.51**	**63.07 / 81.26**	**56.36 / 81.23**	**66.63**	**64.12 / 75.05**
	DNN-AttCP	**62.61 / 81.14**	**59.79 / 79.00**	**62.45 / 80.22**	**54.81 / 80.41**	**64.89**	**63.20 / 74.70**

We first considered the IEMOCAP results in Table 3 to evaluate the different systems in an established benchmark. As it can be clearly noted, in this case the best performance is obtained using deep features from self-supervised acoustic models. Nevertheless, there is no statistical difference between using W2V2 and WavLM as feature extractors when comparing SVM and DNN classifiers. Moreover, the results using the DNN downstream models are comparable with those reported in (Kakouros et al., 2023), which are SOTA metrics in the speech-only benchmark. Thus, the analysis of these results pointed out that the main improvements come from these self-supervised models as feature extractors. At the same time, a simple SVM classifier is robust enough to exploit the paralinguistic information of the deep embeddings to perform SER classification.

Figure 3 shows the confusion matrix for the IEMOCAP dataset using the WavLM feature extractor with DNN-AttCP downstream classifier. It can be observed that the per-class accuracy ranges between 70%-80%, except for the neutral class, which shows the lowest per-class accuracy (68%). Moreover, a high percentage of misclassifications are observed between the neutral and the remaining emotions, which could be expected in this task when the system does not clearly detect the emotion in the speech signal. Indeed, the results are consistent with those obtained by state-of-the-art recent SER studies (Kakouros et al., 2023; Ulgen et al., 2024; Shome and Etemad, 2024).

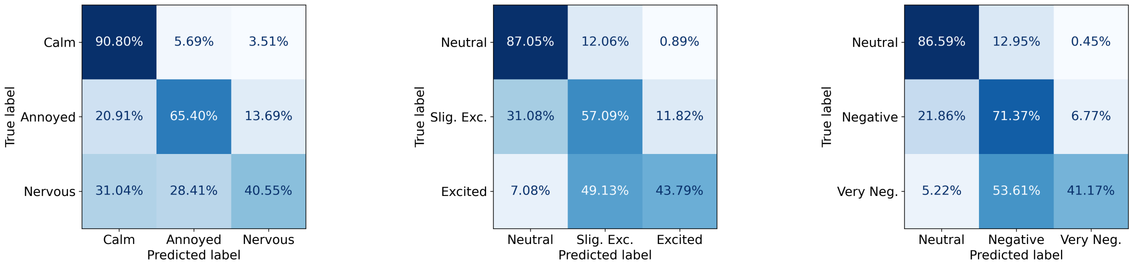
Regarding the in-domain CC datasets, similar tendencies are found. Table 4 shows the experimental results for the CC-Debt and CC-Support datasets. We also included results obtained when using two baseline classifiers: a random classifier (results averaged over 50 trials) and a majority voting classifier. As observed, the best performance is obtained using W2V2 and WavLM features regardless of the classifier. These results confirm the well-known capabilities of self-supervised models for this task. As expected, the label merging process improved different use cases and dimensions, with F1 scores close to suitable values for practical applications (Płaza et al., 2022; Deschamps-Berger et al., 2023). For the case of CC-Debt, the gains obtained for the complete vs. binary level configurations are similar across the different dimensions, showing possible correlations among them.

On the contrary, the disparity is higher in CC-Support. The observed improvements in categorical classification are probably due to the label reduction process. Nevertheless, the F1 metrics are lower in the case of valence and, especially, arousal, where there were two classes from the initial version. The discrepancy with respect to the CC-Debt is that the former is a less-emotional domain, and the data for the minority classes were scarce. Therefore, accurately detecting excitement and negative emotions was even more challenging. Still, the results are competitive, considering the task's complexity (Deschamps-Berger et al., 2023). Other features, such as ComPare16 or x-vectors,



(a) Category (b) Arousal (c) Valence

Figure 4: Confusion matrices for the CC-Support dataset using a W2V2 feature extractor with SVM classifier. These matrices are obtained by dimension (Category, Arousal, Valence) in the complete classes setting.



(a) Category (b) Arousal (c) Valence

Figure 5: Confusion matrices for the CC-Debt dataset using a W2V2 feature extractor with SVM classifier. These matrices are obtained by dimension (Category, Arousal, Valence) in the complete classes setting.

show strong results for particular cases. The linguistic features exhibit higher accuracy in valence prediction within the CC-Support domain, likely attributable to the relative ease of transcription due to the standardized vocabulary prevalent in these conversations.

To further analyze these results, Figures 4 and 5 show the confusion matrices obtained for the CC-Support and CC-Debt datasets, respectively, when evaluating the W2V2 feature extractor with SVM classifier. We only show the results for the complete classes setting as it yields a better understanding of the main errors produced by the systems. For the CC-Support, the detection of emotion categories such as Nervous and Satisfied has a low detection rate due to the few amount of hours, which justifies using a simple binary detection between neutral and emotional classes, with most of the cases representing Annoyed or Nervous users. Similarly, Positive valences are mainly confused with Neutral, while detecting Negative vs others can bring better discriminative results that help detect these altered states. For the CC-Debt, similar behavior is observed for the emotional category detection. Regarding arousal and valence dimensions, extreme classes are mainly confused with the adjacent intermediate level (e.g., Excited with Slightly Excited arousal and Very Negative with Negative valences). Thus, in this scenario, with few labeled data for these classes, it is justified to simplify the detection problem and group the non-neutral levels in a single class while keeping the usefulness of the deployed systems.

In conclusion, using self-supervised features from large speech models with classical machine learning classifiers such as SVMs can obtain promising results for a practical application of SER when considering a simplified scenario focused on detecting Neutral and Emotional classes. Moreover, in more complex scenarios with class variety, the results are still competitive regarding the current state-of-the-art in this area (Kakouros et al., 2023; Deschamps-Berger et al., 2023). It is also important to remark that using general SSL feature extractors trained on a large variety of speech data avoids the need for transfer learning from pretrained SER models (e.g., IEMOCAP), especially when there is a considerable domain shift between the source and target scenarios (different languages, acoustic channel, acted vs. real emotions).

3.3 Subjective analysis

Finally, we conducted a subjective evaluation to further analyze the usability of the SER models in a practical scenario for customer service and speech analytics. To this end, a new test set of 15 recorded conversations from the CC-Debt domain is considered, including both neutral and emotional states. Two different evaluators are involved in the task. The first one listens to the conversations and takes notes about their emotional content and evolution during the call, both for the agent and the client. The second one only analyzes the information provided by the automatic pipeline, which includes segmentation, diarization, and emotion recognition (including both full and binary models). Then, the evaluator described the conversation's emotional evolution using only this information. Finally, the evaluators compared their analysis.

After finalizing this procedure, five conversations were categorized as generally neutral, and the remaining ones as emotional. The analysis of both evaluators matched in 100% of the conversations, regarding general aspects as the evolution of emotional state, considering both agent and client during the conversation. Despite minor errors not only related to the emotional models but also other modules in the pipeline (e.g., speaker segmentation by the diarization step), the automatic analysis allowed us to obtain an overview of the call and evolution of emotional states. Interestingly, the second evaluator remarked on the usefulness of multi-class models for arousal and valence to evaluate the temporal evolution and accurately identify segments with strong negative emotions (excited arousal and very negative valence). On the other hand, using binary models for categorical prediction was preferred to identify the negative state. To summarize the outcomes from this evaluation, the involved company identified practical use cases where the emotional models, along with other speech-related technologies (such as automatic speech recognition and content classification) have potential applicability, including: (1) call identification with very negative emotions from the client (especially exploiting extreme categories for arousal and valence), (2) analysis of emotional evolution on these calls, and (3) evaluation of agents performance, where mid-level emotional states are important to analyze the conversations' emotional evolution.

4 Conclusions

This paper presented a practical technological transfer of speech emotion recognition systems to the CC speech analytics sector. This work results from collaboration between research teams and two CC-related companies to address specific target scenarios. In this study, we completed all the necessary stages to ensure that the systems are production-ready from their facilities: data acquisition, preprocessing and annotation, analysis and design of the experimental framework, training, and evaluation of the different approaches. The CC-expert annotation process ensures the quality of the data while meeting all the privacy concerns, which usually causes several issues in developing these systems. Moreover, we considered two application scenarios to evaluate the original annotated data and a transformed version (2 classes) focused on detecting emotional states. The experimental results indicated that the proposed approaches are competitive for the two scenarios, as well as in a well-established benchmark in the research community. This work represents a successful technological transfer to the industry, where the companies have deployed the solutions to evaluate it in their commercial cases. In future work, we will study these models' use along with active learning techniques to help annotate additional emotional data.

Limitations

The main limitation of this work is the generalization and application of the development of SER systems in out-of-domain conditions. The models have been trained on a limited amount of labeled speech in specific conditions of language, acoustic channel, application domain, and targeted emotions. Thus, using these models under different conditions will result in a performance drop and non-sense results. Therefore, these systems should only be considered under similar conditions.

Another limitation is the dependence of the SER module on previous speech-processing steps in real applications, including speech segmentation and speaker diarization. Thus, errors in the previous steps of the pipeline will ultimately affect the predictions obtained by the SER system.

Ethics Statement

The EU AI Act considers systems that predict human emotions from their biometric data. There is a concern about them due to their potential biases and lack of generalization, as well as their potential to limit rights and freedom for human beings. Thus, these systems are generally considered high-risk and strictly forbidden in domains where specific person profiles are targeted, such as work and education environments. Using these systems in sensitive domains should only be allowed with a healthcare objective or to ensure people's security.

In the context of analyzing call-center conversations, the use of a SER system may be classified as a non-high-risk application, as long as it ensures the protection of individuals' health, security, and human rights. Additionally, it is critical to implement measures that prevent potential biases in the AI models and ensure that these systems do not significantly influence decision-making processes, which should always be reviewed by human experts. Furthermore, this technology must not be used to profile clients within this domain, and the results should always be anonymized to protect individual privacy. Moreover, with regard to transparency obligations, clients should always be informed that their conversations are being recorded and analyzed using these AI systems, and they should be given the right to object to these operations. Finally, the deployment of SER systems in CC scenarios should always be carried out under the supervision of ethics experts to ensure compliance with rules and directives outlined in EU regulations.

Acknowledgements

This work was partially funded by the Basque Business Development Agency, SPRI, under the EMPHASIS project, grant agreement ZE-2021/00039.

References

Arun Babu et al. 2022. XLS-R: Self-supervised cross-lingual speech representation learning at scale. In *Proc. InterSpeech 2022*, pages 2278–2282.

Paul Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot. Int.*, 5(9):341–345.

Milana Bojanić, Vlado Delić, and Alexey Karpov. 2020. Call redistribution for a call center based on speech emotion recognition. *Applied Sciences*, 10(13):4653.

Carlos Busso et al. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. In *PML4DC at ICLR 2020*.

Sanyuan Chen et al. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Mikel de Velasco, Raquel Justo, and María Inés Torres. 2022. Automatic identification of emotional information in spanish tv debates and human–machine interactions. *Applied Sciences*, 12(4):1902.

Théo Deschamps-Berger, Lori Lamel, and Laurence Devillers. 2021. End-to-end speech emotion recognition: challenges of real-life emergency call centers data recordings. In *Proc. 2021 ACII*, pages 1–8.

Theo Deschamps-Berger, Lori Lamel, and Laurence Devillers. 2023. Exploring attention mechanisms for multimodal emotion recognition in an emergency call center corpus. In *Proc. ICASSP 2023*.

Paul Ekman et al. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Florian Eyben et al. 2015. The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202.

Weiquan Fan, Xiangmin Xu, Xiaofen Xing, Weidong Chen, and Dongyan Huang. 2021. LSSED: A large-scale dataset and benchmark for speech emotion recognition. In *Proc. ICASSP 2021*, pages 641–645.

Yajing Feng and Laurence Devillers. 2023. End-to-end continuous speech emotion recognition in real-life customer service call center conversations. In *Proc. 2023 ACII Workshop and Demos*.

Luciana Ferrer and Pablo Riera. Confidence intervals for evaluation in machine learning.

C Hema and Fausto Pedro Garcia Marquez. 2023. Emotional speech recognition using CNN and deep learning techniques. *Applied Acoustics*, 211:109492.

Christian Hildebrand, Fotis Efthymiou, Francesc Busquet, William H Hampton, Donna L Hoffman, and Thomas P Novak. 2020. Voice analytics in business research: Conceptual foundations, acoustic feature extraction, and applications. *Journal of Business Research*, 121:364–374.

Jon Irastorza and M. Ines Torres. 2016. Analyzing the expression of annoyance during phone calls to complaint services. In *Proc. IEEE CogInfoCom*, pages 103–106.

Jon Irastorza and M. Ines Torres. 2019. Tracking the expression of annoyance in call centers. *Cognitive Infocommunications, Theory and Applications*, pages 131–151.

Sofoklis Kakouros, Themos Stafylakis, Ladislav Mošner, and Lukáš Burget. 2023. Speech-based emotion recognition with self-supervised models using attentive channel-wise correlations and label smoothing. In *Proc. ICASSP 2023*.

Mikaela Keller, Samy Bengio, and Siew Wong. 2005. Benchmarking non-parametric statistical tests. *Advances in neural information processing systems*, 18.

Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget. 2022. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Computer Speech & Language*, 71:101254.

Manon Macary, Marie Tahon, Yannick Estève, and Daniel Luzzati. 2023. Acoustic and linguistic representations for speech continuous emotion recognition in call center conversations. *arXiv preprint arXiv:2310.04481*.

Abdelrahman Mohamed et al. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Rosa YG Paccotacya-Yanque, Candy A Huanca-Anquise, Judith Escalante-Calcina, Wilber R Ramos-Lovón, and Álvaro E Cuno-Parari. 2022. A speech corpus of quechua collao for automatic dimensional emotion recognition. *Scientific Data*, 9(1):778.

E Parada-Cabaleiro, G Costantini, A Batliner, A Baird, Bw Schuller, et al. 2018. Categorical vs dimensional perception of italian emotional speech. In *Proc. InterSpeech 2018*, volume 2018, pages 3638–3642.

Luis Felipe Parra-Gallego and Juan Rafael Orozco-Arroyave. 2022. Classification of emotions and evaluation of customer satisfaction from speech in real world acoustic environments. *Digital Signal Processing*, 120:103286.

Leonardo Pepino, Pablo Riera, and Luciana Ferrer. 2021. Emotion recognition from speech using wav2vec 2.0 embeddings. *Proc. InterSpeech 2021*, pages 3400–3404.

Paula Andrea Pérez-Toro, Juan Camilo Vásquez-Correa, Tobias Bocklet, Elmar Nöth, and Juan Rafael Orozco-Arroyave. 2021. User state modeling based on the arousal-valence plane: Applications in customer satisfaction and health-care. *IEEE Transactions on Affective Computing*, 14(2):1533–1546.

Mirosław Płaza et al. 2022. Emotion recognition method for call/contact centre systems. *Applied Sciences*, 12(21):10951.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*, pages 28492–28518.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Debaditya Shome and Ali Etemad. 2024. Speech emotion recognition with distilled prosodic and linguistic affect representations. In *Proc. ICASSP 2024*, pages 11976–11980.

Themos Stafylakis, Ladislav Mošner, Sofoklis Kakouros, Oldřich Plchot, Lukáš Burget, and Jan Ćernockỳ. 2023. Extracting speaker and emotion information from self-supervised speech models via channel-wise correlations. In *Proc. IEEE SLT 2022*, pages 1136–1143.

Szabolcs Levente Tóth, David Sztahó, and Klára Vicsi. 2008. Speech emotion perception by human and machine. In *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction: COST Action 2102 International Conference*, pages 213–224.

Ismail Rasim Ulgen, Zongyang Du, Carlos Busso, and Berrak Sisman. 2024. Revealing emotional clusters in speaker embeddings: A contrastive learning strategy for speech emotion recognition. In *Proc. ICASSP 2024*, pages 12081–12085.

Mikel deVelasco Vázquez, Raquel Justo, Asier López Zorrilla, and Marıa Inés Torres. 2019. Can spontaneous emotions be detected from speech on tv political debates? In *10th IEEE International Conference on Cognitive Infocommunications*, page 289.

Shu wen Yang et al. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. InterSpeech 2021*, pages 1194–1198.

Fangfang Zhu-Zhou, Roberto Gil-Pita, Joaquín García-Gómez, and Manuel Rosa-Zurera. 2022. Robust multi-scenario speech-based emotion recognition system. *Sensors*, 22(6):2343.

Probing Whisper Predictions for French, English and Persian Transcriptions

Nicolas Ballier[1], Léa Burin[2], Behnoosh Namdarzadeh[2], Sara Ng[3], Richard Wright[3], Jean-Baptiste Yunès[4]
[1]LLF & CLILLAC-ARP, Université Paris Cité, F-75013 Paris, France
[2]CLILLAC-ARP, Université Paris Cité, F-75013 Paris, France
[3]University of Washington, USA
[4]IRIF, Université Paris Cité, F-75013 Paris, France
Contact: `nicolas.ballier@u-paris.fr`

Abstract

Whisper is a widely-used open-access Large Language Model (LLM) trained using a multilingual paradigm. As such it represents an important opportunity for researchers to study how multilingual LLMs function across languages. In this paper, we analyse Whisper's Large and Medium models for Persian, English and French using a transcription task. To investigate the calibration of Whisper models, we use a customised C++ version of Whisper to probe Whisper's internal representations by extracting the subtoken probabilities for transcriptions of speech samples of the target languages. We discuss our subtoken-based evaluation of prediction accuracy as a proxy for standard Word Error Rate evaluation of the different Whisper models. The accuracy of the ASR predictions is investigated as a function of target language and part of speech. Our analysis reveals an architectural bias for French and discrepancies in accuracy in relation to the size of the training data. The results of our novel subtoken-based evaluation supplement previously-reported cross-lingual evaluations of Whisper, and enable better fine-tuning by suggesting types of data that may improve calibration.

1 Introduction

Large Language Models (LLMs) are still perceived as black boxes. Recent papers have mostly described new state-of-the-art performance on transcription tasks with LLMs, but the reliability of different implementations has not, to the best of our knowledge, been investigated on the basis of the probability of the subtokens predicted by the LLMs. It should be borne in mind that LLMs do not predict tokens but subwords or subtokens, as the result of the byte pair encoding (BPE) (Sennrich et al., 2016), a compression algorithm adapted from Gage (1994). Taking advantage of its publicly available models and of the C++ implementation (Gerganov, 2003), whisper.cpp (hereafter "Whisper"), we probed the Whisper system and retroconverted the timestamps into a TextGrid (see Figure 1) in order to inspect the speech data. Our reverse engineering strategy is illustrated in Figure 1. We extracted timestamps and subtoken probability for each subtoken prediction.

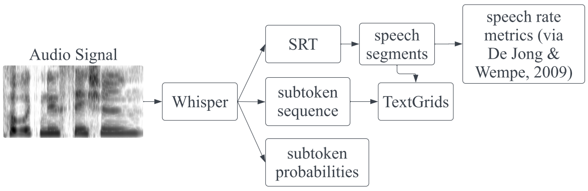


Figure 1: Extracting information from the Whisper pipeline (Radford et al., 2023) with Gerganov (2003).

Though Whisper has been trained with 680,000 hours of audio, out of which 117,000 hours represent 96 other languages than English, the distribution of the training data is heavily skewed, as indicated in the appendix of Radford et al. (2023)[1]. For inclusive spoken language science and technology, this discrepancy in the training data, as illustrated in Table 1, has a price worth investigation. We will show that Whisper's Large model has different calibration curves for Persian, French and English. The rest of the paper is structured as follows: Section 2 summarises previous research on Whisper and contextualises our contribution in this respect. Section 3 presents our experiment

Table 1: Number of hours of Whisper's training data for French, Persian and English, after Radford et al. (2023).

Speech Recognition	**Translation**
French : 9,752	French : 4,481
Persian : 24	Persian : 302
English : 563,000	

---

[1] Since "Of those 680,000 hours of audio, 117,000 hours cover 96 other languages" we assumed that English was trained with 563,000 hours (680,000-117,000).

design, including the data and methodology. Section 4 presents the results. In Section 5 we discuss them. Section 6 concludes the paper.

2 Previous Research

Whisper is an audio Large Language model that has been trained for several tasks such as Voice Activity Detection, Transcription, textual translation into English and language detection (Radford et al., 2023). Less than two years after the public release of its models, more than 1,900 papers have been written using Whisper according to Google scholar. Many researchers have tried to optimise Whisper, for example by post-processing Whisper's outputs with LlaMa (Touvron et al., 2023) in a framework (Radhakrishnan et al., 2023) or to integrate Whisper in robots (Pande and Mishra, 2023). Whisper has been trained with LibriSpeech (Panayotov et al., 2015) data, whose features have been investigated for speech synthesis (Zen et al., 2019; Kakouros et al., 2023). Whisper has been tested with the FLEURS dataset (Conneau et al., 2023) for the Spoken Language Identification (SLID) task (Augenstein and Salaj, 2023) and used for Spoken Language Understanding (SLU) (Wang et al., 2023). Whisper has also been used for deep fake analysis (Kawa et al., 2023) and the whisper.cpp (Gerganov, 2003) implementation has been used to score second language speech (Ballier et al., 2023a). Coupled with an SVM classifier, Whisper showed good results when classifying vocal intensity categories (soft, normal, loud, and very loud) from speech signals (Kodali et al., 2023). Sun et al. (2023) have tested biasing lists to improve Whisper's speech recognition, which has also been improved when piped to the LLM LlaMa (Touvron et al., 2023) to select Whisper ASR outputs (Radhakrishnan et al., 2023). Analysing different varieties of English, Graham and Roll (2024) showed that Whisper performance was better for read speech than for spontaneous speech. They also showed that performance for Canadian and American English was comparable, but it was poorer for British and Australian English. Previous research on Whisper outputs has shown that the different segments produced by the different models are not identical in numbers and scope and differ from the speech signal (Ballier et al., 2023b). Several probing methods have been applied to LLMs, for example probing prompts (Qi et al., 2023), but Whisper probability distributions have not been investigated, to the best of our knowledge. The closest work to our research is a previous attempt to understand the information flux for the plural agreement in French, using a forced aligner and attention heatmaps, showing that agreement is dealt with in Whisper by the decoder, not the encoder of the Whisper `Transformer` architecture (Mohebbi et al., 2023).

3 Material and Methods

One previous study (Ballier et al., 2023b) suggests that when the Whisper outputs are not normalised (contrary to the normalisation procedure used in Radford et al. (2023)'s benchmarks and described in its appendix), word error rate (WER) is lower for the medium model than for the large model. We wanted to investigate the accuracy of the two models, as well as investigate how degraded the performance is when the training data size decreases. To this aim, we resorted to the calibration curve method, that plots the probability assigned to the subtoken (x) on the accuracy of its prediction (y axis). Best calibrated models are close to the $x = y$ axis and overconfident models are much above this axis. We present the calibration curve method before specifying the data we used for our tests.

3.1 The Calibration Curve Method

Assuming the probability assigned by the system to predicted subtokens is a correlate of confidence, we believe that for trustworthy AI we should investigate subtoken probabilities, especially when the prediction is wrong. A method to achieve this reliability analysis is the "calibration curve" method. This method has been used to analyse neural networks (Guo et al., 2017; Minderer et al., 2021) and, recently, to assess LLMs from a semantic point of view. For instance, Levinstein and Herrmann (2024) use calibration curves to assess the truthfulness of LLM statements on specific datasets and claims that "calibration provides another metric for evaluating the quality of the probes' forecasts"[2]. Calibration allows researchers to examine whether the model predictions are on average too certain (overconfident) or too uncertain (underconfident) (Minderer et al., 2021), this paving the way for LLM recalibration (Chen et al., 2024). Because our analysis is based on subtokens, we also computed a regression model to assess the role of subtokenisation, fitting a logistic model with the number of

---

[2] The concept was used initially to analyse the reliability of weather forecasts (Brier, 1950; DeGroot and Fienberg, 1983).

subtokens as one of the predictors.

3.2 Logistic Regression Modelling

We fitted a logistic regression model with accuracy (success rate) of the Whisper predicted subtokens as the predicted variable and several variables for predictors. We tested duration, segment, speaker, overlaps, speech rate and phonation rate. We extracted the turns from the official transcripts of the corpus (Branca-Rosoff, Sonia, 2013). We also computed the number of subtokens required to represent a token in the final transcription and POS-annotated the corresponding token. We used one of the treebanks for English to annotate the data from the ATAROS corpus (Freeman et al., 2014). We used the EWT model for the universal dependency annotation, based on English Web Treebank corpus (Silveira et al., 2014).

3.3 Data

We used challenging data to test the ASR task, since the LLM was trained with read speech from Librispeech (Panayotov et al., 2015).

- **Persian** We used two recordings from two female Tehrani speakers reading 120 sentences containing a dislocation. The sentences, extracted from various sources, reflect a formal register. Each sentence encapsulates the linguistic phenomenon of dislocation. Speakers recorded their voices on Zoom while reading each sentence aloud, since Whisper was trained to deal with noisy environments (Radford et al., 2023). We avoided overlaps and spontaneous speech because of the number of hallucinations observed when transcribing Persian.

- **French** For French, we used almost one hour (55 min.) of spontaneous French collected for the CFPP reference corpus (corpus de français parlé parisien) (Branca-Rosoff et al., 2009). This conversation of a dyad was collected in the early 2000's in Paris and has already been scrutinised from a prosodic perspective (Martin, 2020; Morel, 2011; Cresti et al., 2011).

- **English** For English, we used the ATAROS corpus (Freeman et al., 2014), designed to investigate stance and engagement in collaborative tasks. This corpus consists of dyadic conversations between unfamiliar interlocutors. Dyads of native English speakers from the Pacific Northwest of the United States (unknown to each other but roughly matched for age) completed a variety of collaborative tasks (Freeman, 2015). We present results from 2 sessions (56 min.) of mixed-gender dyads. We apply a temporal filter to the audio based on timings from human transcriptions of the target speaker, to mitigate non-target speech.

3.4 Data Extraction and Processing

We applied the following pipeline to our data:

- For reference corpora, we extracted the timestamps delimiting turns to create a speaker variable, and an overlap variable;

- With whisper.cpp, we extracted timestamps, subtoken predictions and the probability associated with each prediction;

- We qualitatively annotated the prediction of the Whisper model, assigning 0 to error and 1 to predictions. We report accuracy (success rate) and do not take into account omissions or word error rate (WER) because our analysis is at a subtoken level (we discuss the implications in relation to standard ASR based on WER in subsection 5.4);

- From whisper.cpp, we extracted the .SRT files that gave the timestamps of the segments created by the different Whisper models;

- With a series of scripts we computed the speech rate for each segment.

Using the C++ Whisper implementation (whisper.cpp), we also retrieved timestamps aligned to the Whisper segments, encapsulated in the extracted .SRT files. We then extracted the segment ID timestamps from the .SRT file, and mapped them onto the 16,131 prediction timestamps. We associated the 1,415 segments to their speech rates, computed with the De Jong and Wempe Praat plugin (De Jong and Wempe, 2009). We then checked for overlaps using the official transcript of the corpus (Branca-Rosoff, Sonia, 2013). Using the .trs (xml) file, we coded Whisper subtoken predictions corresponding to overlaps.

By default, we considered that we had no gold standard reference for the special tokens, so we discarded the special tokens (which we assumed to be correct predictions by default) as well as punctuation. Homophones were counted as errors as they did not semantically match the reference transcription '*voir*' vs '*voire*'; '*m'aime*' vs '*même*'; '*ah c'est bon*' vs '*assez bons*'. Because Whisper was assessed with a normalisation procedure, we counted as correct '*17 and a half*' when the reference transcription had '*17,5*'.

3.5 A Brief Presentation of Whisper Byte-Pair-Encoding

For all languages and models, Whisper codes linguistic input as a composition of 51,866 subtokens. We provide a series of examples of the different types of tokens acknowledged in the HuggingFace documentation[3] and which can be explored in the dictionary of subtokens. We indicate the subtoken ID (number) which we found in the dictionary of subtokens.

- 50,255 linguistic subtokens, corresponding to English words or fragments for French or graphemes for languages like Persian;

- special tokens, some of them corresponding to boundaries of the `Transformer`: the end of text and end of sentence subtokens 50257 *[_EOT_]* and 50258 *[_SOT_]*;

- 100 extra-tokens labelled *[_extra_token_50259]* to *[_extra_token_50359]*;

- 7 special tokens are also acknowledged in the literature such as 50360 *[_SOLM_]*, 50361 *[_PREV_]*, 50362 *[_NOSP_]*, 50363 *[_NOT_]* and 50364 *[_BEG_]*. *[_BEG_]* corresponds to the beginning of the 30 second window when the sound file is processed by Whisper;

- 1,500 out-of-vocabulary OOV subtokens from *[_TT_1]* to *[_TT_1500]*. We will show that they correspond to temporal subtokens and we examine their status in subsection 4.2.

Our pipelines to investigate the Whisper inner computations is available on the GitHub of the sixth author.[4] We created automated scripts with R for the transformation of Whisper outputs into Praat TextGrids.

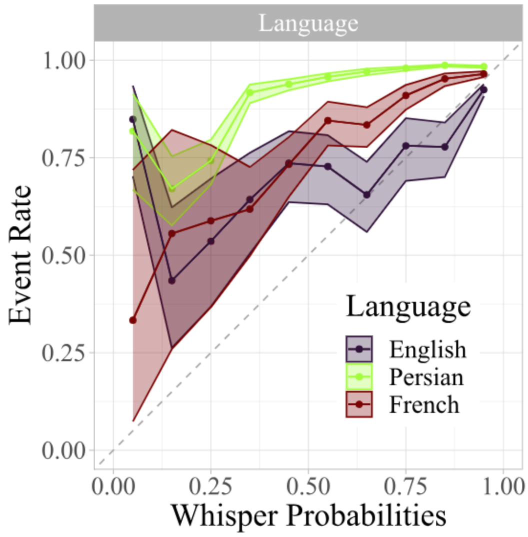


Figure 2: Calibration Curve for the Transcription of English, French, and Persian.

4 Results

4.1 Language Effects

We compared the calibration curves for the large models for the transcription of the three languages. Figure 2 shows the overconfidence of the Whisper model for Persian and French, well above the x=y line corresponding to the ideal calibration. When transcribing English, the predictions of the large model only partially overlap with ideal calibration.

4.2 Whisper's Internal Correlates to Temporal Values

One of the in-built limitations of the Whisper architecture is that audio inputs are limited to 30 second segments. When audio duration is greater than 30 seconds, the model must additionally truncate the audio at intermediate intervals. It appears that the so-called *TT* tokens may be outputs from this process. We analysed the main outputs of out of vocabulary *TT* tokens predicted by Whisper every time a punctuation symbol was used. We also analysed the property of the out of vocabulary tokens, the special tokens corresponding to end of text, end of sentence, and BEG, which structures the windowing of Whisper. In this subsection, we present the different types of results we obtained based on Whisper medium outputs on the Inventory

---

[3] https://huggingface.co/docs/transformers/model_doc/whisper

[4] https://github.com/jbyunes/whisper.cpp

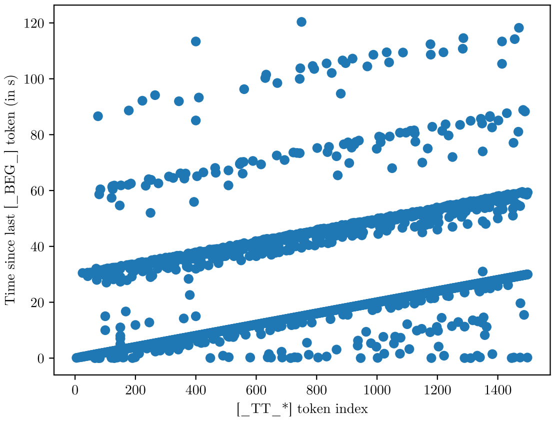


Figure 3: Comparision between the indices of *[_TT_*]* tokens and the time since the previous *[_BEG_]* token, across audio data from the ATAROS corpus.

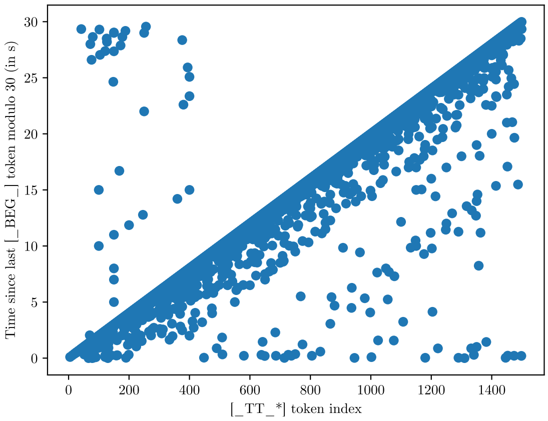


Figure 4: Comparison between the indices of *[_TT_*]* tokens and modulated spurt time, across audio data from the ATAROS corpus.

and Budget subtasks of the ATAROS dataset (Freeman et al., 2014). Based on the hypothesis that the numeric index of the *[_TT_*]* tokens were approximate to 20ms intervals since start of an audio span, we conducted a linear regression between the index of all *[_TT_*]* tokens in the output and the reported time difference between the token and the nearest *[BEG]* token in the previous output. Based on the observation that in some regions of the output the *[_TT_*]* token indices seem to "reset" without an intervening [BEG] token, we also conduct a linear regression between the *[_TT_*]* token indices and the time since the previous *[_BEG_]* token ***modulo 30s***. Figure 3 compares the token indices to the time since the previous *[_BEG_]* token, and Figure 4 compares the token indices to the modulated time since the previous *[_BEG_]* token. The regressions for both settings were significant ($p < 1e - 15$). The r-value for the regression between the token indices and the raw time since the *[_BEG_]* token was 0.771, and when comparing to modulated time the r-value was 0.990.

4.3 Architectural Bias

Because Whisper predicts subtokens, not tokens, after the byte pair encoding (Sennrich et al., 2016), we created a subtoken_cnt variable corresponding to the number of subtokens needed to represent a given token. Previous research on neural machine translation has shown that gender bias for French into English translations can be sensitive to the number of subtokens required to represent tokens referring to female occupational nouns. We observed a similar architectural bias for French since the accuracy decreases with the number of subtokens, as can be seen on Figure 5. More research is needed to analyse how this might be a confounding factor for the mistranscription of named entities. The architectural bias was not observed in the regression model for Persian but it may be because the dataset was too small. Architectural bias (if not training bias) was observed in the accuracy of the prediction of the subtokens, which was distributed unevenly. Some subtokens were systematically accurate or wrong in the transcriptions, independently of the estimated probability as can be seen on Fig-

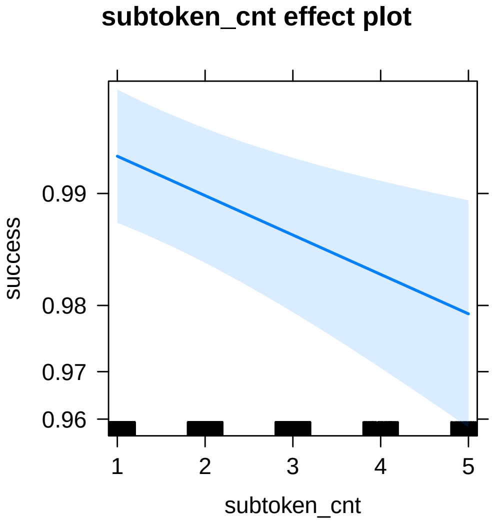


Figure 5: Effect of the number of subtokens on the transcription task for French (Large model).

ure 6.[5]

4.4 POS Effect

Universal dependency annotation provides two types of part-of-speech annotation, one which is more general and follows universal guidelines to describe categories that are posited to be universally valid for all languages (upos). Other language-related tagsets entrenched in the metalinguistic tradition of a given language (xpos) are also acknowledged, as is the case here for English with the Penn Treebank tagset displayed in Table 2. The default tagset (WET) used to analyze English was trained on web data, so it was not particularly designed to annotate spoken corpora. Some observations can still be made, some part-of-speech categories being more prone to phonetic variation such as *to* (xpos: TO) and determiners (xpos:DT), which are more likely to undergo alternations between reduced and full vowels, as can be seen in Table 2. If we take into account the universal part-of-speech (upos) tagset, perfect success rates can be achieved for categories such as conjunctions of coordination. On the other hand, weak forms or determiners are likely to undergo more ambiguous transcriptions probably because of the weak forms of *a* and other determiners. Similarly, weak forms may account for the rather poor score for auxiliaries and pronouns. There is thus an effect of weak forms and their potential ambiguities.

Table 2: Best categories predicted.

xpos	n	success
TO	39	0.82
DT	102	0.90
PRP	68	0.94
WDT	15	1
CD	17	1
CC	66	1
IN	77	1

upos	n	success
AUX	71	0.95
PRON	127	0.96
NOUN	172	0.97
NUM	17	1
SCONJ	48	1
ADP	56	1
CCONJ	66	1

5 Discussion

5.1 Suggestions for Fine-tuning Whisper for Persian

It seems that the models for Persian are less robust, as a very strong inter-speaker variability can be observed. For the speech recognition of the same sentences, the success rate varies from 95% to 55% and this speaker effect can be seen on the calibration curves on Figure 7. Overfitting of the Persian model with Arabic data needs to be stressed as well. This can be explained by a partly commonly shared alphabet between the two languages. Persian has a few specific graphemes for the voiced velar and significantly more homophones than in Arabic. Nevertheless, some very specific Arabic letters are used instead of Persian like the nasal consonant, the alveolar nasal and other substitutions can be observed. Furthermore, the Perso-Arabic script used to write Persian is cursive, meaning that letters tend to have different shapes depending on whether they join with adjacent letters or not. The different graphotactics of Persian for initial, medial, and final characters are not represented in the sub-tokenization of Persian transcription. Considering the phonotactic and graphotactic constraints of Persian showcased in the transcription by Whisper, fine-tuning Whisper could be a way to improve the transcriptions of a language with low training data.

5.2 The Locus of Hallucinations

In the case of Persian, smaller models of Whisper exhibited some hallucinations, which can be attributed to the subtoken dictionary. However, these hallucinations were not present in the larger model. The occurrence of hallucinations is not consistent across different models. Specifically, in smaller models like the `small` model, numbers read by the speaker at the beginning of each sentence were often hallucinated. For English and French, we mostly observed “coda" hallucinations as in Figure 8. Within the two seconds after the end of speech intervals, transcriptions are provided in spite of the absence of speech signal. Our hypothesis is this comes from the training data (probably from

---

[5] This subtoken-by-subtoken analysis could be replicated with English and French but with more difficulty, as the order magnitude for the number of subtokens is 1 to 8 for Persian to French and 1 to 40 for Persian to English, an estimation based on Google’s Compact Language Detector 3.

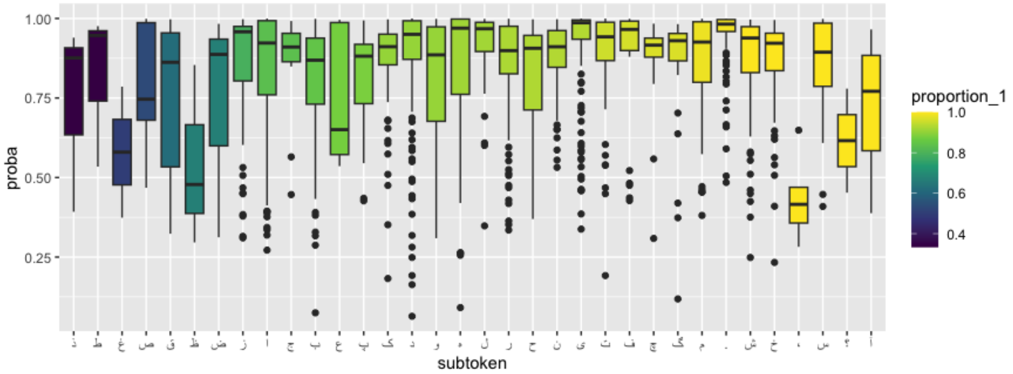


Figure 6: Distribution of probability for Persian subtokens of one character.

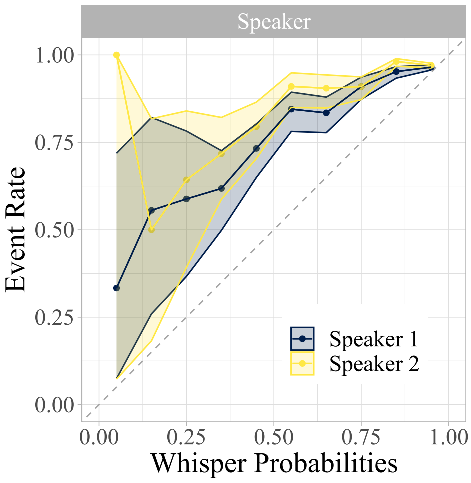
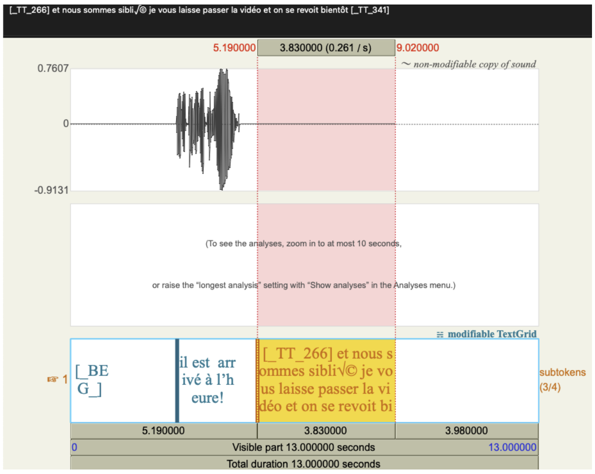


Figure 7: Speaker effect on the quality of the prediction in relation to the confidence of the model (Persian data, Large model).

YouTube) which contains final formulae like “see you soon", here “je vous laisse la vidéo et on se voit bientôt" (“I’ll leave you the video and see you soon."). For English we often found “Thank you" in the coda hallucinations.

5.3 Reliability of Whisper’s Timestamps?

Many special tokens separating subtokens have no duration and some .SRT files are uneasily retro-converted to TextGrids. Moreover, the timings do not match up very well with the word-level timings for ATAROS, which is why we reported two estimations for overlap labels – one based on Whisper’s timing, and one version based on the human-aligned timings. Figure 9 shows the discrepancies of duration according to the two methods, whether for words or subtokens.



[_TT_266] et nous sommes sibli√© je vous laisse passer la vidéo et on se revoit bientôt [_TT_341]

(To see the analyses, zoom in to at most 10 seconds,

or raise the “longest analysis” setting with “Show analyses” in the Analyses menu.)

5.190000
3.830000 (0.261 / s)
9.020000
0.7607
0
-0.9131
~ non-modifiable copy of sound
modifiable TextGrid
1
[_BEG_]
il est arrivé à l’heure!
[_TT_266] et nous sommes sibli√© je vous laisse passer la vidéo et on se revoit bi
subtokens (3/4)
5.190000
3.830000
3.980000
0
Visible part 13.000000 seconds
13.000000
Total duration 13.000000 seconds

Figure 8: Coda hallucination in French. The hallucination disappears with the same Whisper model if the pause after the utterance is reduced.

5.4 The Censorship of Repetitions

Our analysis of the success rate is a precision analysis rather than an analysis of recall. We based our analysis on the Whisper predictions, not on the official transcripts of the corpora when available. For English, we also computed an analysis of recall, namely comparing the Whisper predictions to the original transcription of the ATAROS data. As part of the discussion, we computed the difference between using reference text to the corpus as the baseline to which we annotated the prediction of the Whisper models, and we compared this method with the raw output of the Whisper models that was annotated only on the basis of the predictions. Using the first method, we report a 79% success rate, and then we re-aligned only the prediction of the LLMs and computed the success rate. In our accuracy-based analysis, the omissions from the scripts, and in particular all the censorship of the repetitions of the data, were more favourable to the

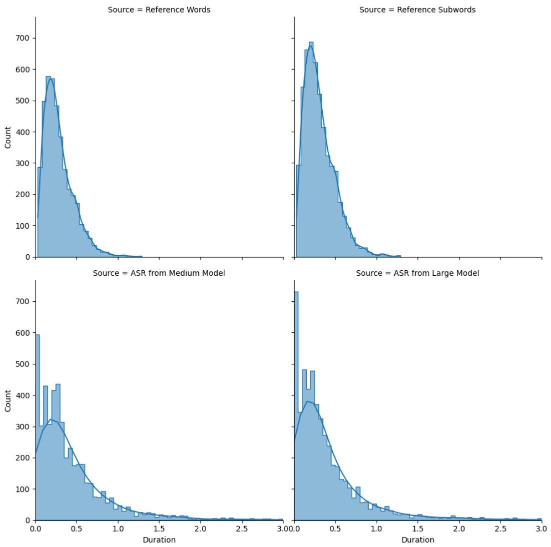


Figure 9: Distribution of duration according to the ATAROS reference transcription (top) and according to Whisper's Large model (ASR, bottom)

interpretation of the Whisper success rates, since we achieved 89% of success using this methodology based on the analysis of the Whisper output only. Among the Whisper omissions in the transcriptions, repetitions accounted for 11.9% of the omitted tokens. Numbers (and generally speaking, counting) accounted for 37.5% of the omissions from the reference text.

6 Conclusions

In this paper, we have probed Whisper outputs using a C++ implementation of Whisper (Gerganov, 2003) to probe the accuracy of transcriptions on a subtoken basis. We use subtoken probabilities and internally produced timestamps. We used reverse engineering to translate the inner working of a large language model, namely its prediction properties, to realign them to the speech signal with Praat (Boersma, 2024) TextGrids. Our method suggests coherent meanings to the special temporal subtokens *[_TT_*]* used by Whisper. This type of research aims to contribute to the explainability of LLMs. The same method could be applied to the translation task; we have here investigated the probabilities associated with the subtokens produced by Whisper for the transcription task. Estimating LLM ASR output at subtokens level allows us to access transcriptions at a finer-grained level and it paves the way for other analyses currently used in the semantic analysis of LLMs such as grouping loss (Perez-Lebel et al., 2023). It should also be noted that analysing subtokens is another way to ensure hallucination detection: subtokens representing Arabic or Japanese were observed for Persian. An unexpected finding is that Whisper scores only report the Persian letters in their isolated forms (abstract representation) and positional variants of letters as observed in the Whisper textual transcriptions seem to be the result of some post-processing. Further studies are needed to investigate this point.

We have shown the effect of size in the training effect, but also an architectural bias for French. It would be interesting to apply the same methodology to explore the probabilities assigned to the translation task to confirm these biases and effects. Analysing the Whisper performances on other languages may confirm one of our observations. With the R (R Core Team, 2024) package Calibratr (Schwarz and Heider, 2019), we also computed the Expected Calibration Error (ECE), which returns the maximum calibration error for equal-frequency binning model (Naeini et al., 2015) for the transcriptions (large model) of the three languages. With the proviso that we have only analysed the transcriptions of three languages with Whisper, a linear model can be fitted with the log of the size of the training data (adjusted R-square 0.99) and it may be the case that the ECE is inversely proportional to the log of the size of the training data, as can be observed on Figure 10.

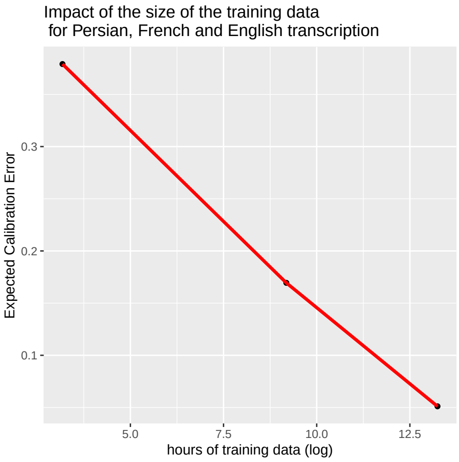


Figure 10: According to Whisper's Large model (ASR task), the effect of the size of the training data

Limitations

As for Persian, our test set consists of read speech focusing on a linguistic construction, dislocation.

We have not used the fine-tuned XLSR-53 large model for speech recognition in Persian (Grosman, 2021) in this study. Using the train and validation splits of Common Voice 6.1 in this fine-tuned model may change the results.

References

Benedikt Augenstein and Darjan Salaj. 2023. Exploiting foundation models for spoken language identification. In *Proceedings of the LWDA 2023 Workshops: BIA, DB, IR, KDML and WM*, pages 28–40.

Nicolas Ballier, Adrien Méli, Maelle Amand, and Jean-Baptiste Yunès. 2023a. Using whisper LLM for automatic phonetic diagnosis of L2 speech, a case study with French learners of English. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 282–292, Online. Association for Computational Linguistics.

Nicolas Ballier, Behnoosh Namdarzadeh, Maria Zimina, and Jean-Baptiste Yunès. 2023b. Translating dislocations or parentheticals : Investigating the role of prosodic boundaries for spoken language translation of French into English. In *Proceedings of Machine Translation Summit XIX: Users Track*, pages 119–132, Virtual. Association for Machine Translation in the Americas.

Paul Boersma. 2024. Praat: doing phonetics by computer. *http://www. praat. org/.*

Sonia Branca-Rosoff, Serge Fleury, Florence Lefeuvre, and Mat Pires. 2009. Discours sur la ville. corpus de français parlé parisien des années 2000.

Branca-Rosoff, Sonia. 2013. Entretien de anita musso (annotations).

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Lihu Chen, Alexandre Perez-Lebel, Fabian M Suchanek, and Gaël Varoquaux. 2024. Reconfidencing LLMs from the Grouping Loss Perspective. *arXiv preprint arXiv:2402.04957*.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Emanuela Cresti, Massimo Moneglia, and Ida Tucci. 2011. Annotation de l'entretien d'Anita Musso selon la théorie de la langue en acte. *Langue française*, 170(2):95–110.

Nivja H De Jong and Ton Wempe. 2009. Praat script to detect syllable nuclei and measure speech rate automatically. *Behavior research methods*, 41(2):385–390.

Morris H DeGroot and Stephen E Fienberg. 1983. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.

Valerie Freeman. 2015. *The Phonetics of Stance-taking*. Ph.d. thesis, University of Washington, Seattle, USA.

Valerie Freeman, Julian Chan, Gina-Anne Levow, Richard Wright, Mari Ostendorf, and Victoria Zayats. 2014. Manipulating stance and involvement using collaborative tasks: an exploratory comparison. In *Proc. Interspeech 2014*, pages 303–307.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Georgi Gerganov. 2003. whisper.cpp : A high-performance inference of OpenAI's whisper automatic speech recognition (ASR) model. *https://github.com/ggerganov/whisper.cpp*.

Calbert Graham and Nathan Roll. 2024. Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. *JASA Express Letters*, 4(2).

Jonatas Grosman. 2021. Fine-tuned XLSR-53 large model for speech recognition in Persian. `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-persian`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Sofoklis Kakouros, Juraj Šimko, Martti Vainio, and Antti Suni. 2023. Investigating the Utility of Surprisal from Large Language Models for Speech Synthesis Prosody. In *Proc. 12th ISCA Speech Synthesis Workshop (SSW2023)*, pages 127–133.

Piotr Kawa, Marcin Plata, Michał Czuba, Piotr Szymański, and Piotr Syga. 2023. Improved DeepFake Detection Using Whisper Features. In *Proc. INTERSPEECH 2023*, pages 4009–4013.

Manila Kodali, Sudarsana Reddy Kadiri, and Paavo Alku. 2023. Classification of Vocal Intensity Category from Speech using the Wav2vec2 and Whisper Embeddings. In *Proc. INTERSPEECH 2023*, pages 4134–4138.

Benjamin A Levinstein and Daniel A Herrmann. 2024. Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*, pages 1–27.

Philippe Martin. 2020. L'annotation prosodique dans ORFÉO. *Langages*, 219(3):103–115.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694.

Hosein Mohebbi, Grzegorz Chrupała, Willem Zuidema, and Afra Alishahi. 2023. Homophone disambiguation reveals patterns of context mixing in speech transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8249–8260.

Mary-Annick Morel. 2011. Les paragraphes intonatifs d'Anita Musso: entre consensus coénonciatif et égocentrage colocutif. *Langue française*, 170(2):111–126.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, page 2901–2907.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Akshara Pande and Deepti Mishra. 2023. The synergy between a humanoid robot and whisper: Bridging a gap in education. *Electronics*, 12(19).

Alexandre Perez-Lebel, Marine Le Morvan, and Gaël Varoquaux. 2023. Beyond calibration: estimating the grouping loss of modern neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Shuhan Qi, Zhengying Cao, Jun Rao, Lei Wang, Jing Xiao, and Xuan Wang. 2023. What is the limitation of multimodal llms? a deeper look into multimodal llms through prompt probing. *Information Processing & Management*, 60(6):103510.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Srijith Radhakrishnan, Chao-Han Yang, Sumeer Khan, Rohit Kumar, Narsis Kiani, David Gomez-Cabrero, and Jesper Tegnér. 2023. Whispering LLaMA: A Cross-Modal Generative Error Correction Framework for Speech Recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10007–10016.

Johanna Schwarz and Dominik Heider. 2019. GUESS: Projecting machine learning scores to well-calibrated probability estimates for clinical decision making. *Bioinformatics*, 35(14):2458–2465.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Natalia Silveira, Timothy Dozat, Marie-Catherine De Marneffe, Samuel R Bowman, Miriam Connor, John Bauer, and Christopher D Manning. 2014. A gold standard dependency corpus for English. In *LREC*, pages 2897–2904. Citeseer.

Guangzhi Sun, Xianrui Zheng, Chao Zhang, and Philip C. Woodland. 2023. Can Contextual Biasing Remain Effective with Whisper and GPT-2? In *Proc. INTERSPEECH 2023*, pages 1289–1293.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMa: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Minghan Wang, Yinglu Li, Jiaxin Guo, Xiaosong Qiao, Zongyao Li, Hengchao Shang, Daimeng Wei, Shimin Tao, Min Zhang, and Hao Yang. 2023. WhiSLU: End-to-End Spoken Language Understanding with Whisper. In *Proc. INTERSPEECH 2023*, pages 770–774.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.

Sawaal: A Framework for Automatic Question Generation in Urdu

Maria Rahim and **Shakeel Khoja**
School of Mathematics and Computer Science
Institute of Business Administration
Karachi, Pakistan
{mrkhowaja, skhoja}@iba.edu.pk

Abstract

This study proposes a novel framework for automatic question generation (AQG) designed specifically for the Urdu language. The framework encompasses seven stages, including pre-processing, tagging, anaphora resolution, word chunking, automatic constructed dataset development (ACD) using Urdu linguistic rules, fine-tuning MT5 on various model combinations, and a ranking algorithm. It includes novel algorithms for anaphora resolution and word chunking customized for Urdu's intricate sentence structures. Utilizing these linguistic rules, the framework generates 4,497 question-answer pairs from 250 passages. Additionally, the framework explores fine-tuning Multilingual T5 (MT5) on UQuAD 1.0 (Kazi and Khoja, 2021) and ACD with varied encodings and embeddings, both with and without the proposed algorithms. Generated questions undergo a ranking process based on semantic text representation to ensure relevance and quality, filtering out irrelevant questions. Evaluation using standard automatic metrics such as BLEU-4, METEOR, and ROUGE-L demonstrates the framework's efficacy, with the best-performing model achieving commendable scores of 24.78, 37.07, and 54.99, respectively.

1 Introduction

Automatic question generation (AQG) is an innovative technique that uses artificial intelligence and natural language processing to create questions from textual information. By employing algorithms and language models, AQG can analyze text, understand context, identify key information, and formulate relevant questions. This technology is useful in various domains such as education (Laban et al., 2022), e-learning, content creation, and chatbot systems.

Despite significant advancements in AQG technology for languages like English, there remains a notable research gap in developing an AQG framework specifically for the Urdu language. Multilingual models often fail to capture the unique linguistic, syntactic, and cultural characteristics of individual languages, especially those underrepresented in training datasets, such as Urdu. Urdu's complexity, with characters forming different shapes based on their position in a word and context-sensitive grammar, adds to the challenge (Daud et al., 2017). For instance, for "how many" questions, there can be a variant of کتنے,کتنا,کتنی,کن depending upon the context and grammar of the question. Moreover, the placement of question keywords can change the meaning of the question. For instance, the keyword "کیا" can be used to form yes/no or what questions depending upon its placement in the question. If "کیا" is placed at the start of the sentence and follows the structure of کیا + subject + verb + helping verb, it will generate yes/no questions, for example, (Does it taste good?) “کیا اس کا ذائقہ اچھا ہے؟" . On other hand, if "کیا" is placed in the middle of the sentence it will form a what question, for example, (What is your name?) "آپ کا نام کیا ہے؟".

The absence of comprehensive frameworks and resources for Urdu limits the availability of automated question generation tools tailored to the specific needs of Urdu-speaking learners, educators, and researchers. To address this research gap, the study proposes a hybrid framework for automatic question generation that combines the linguistic of Urdu with multilingual transformer to generate questions automatically. The major research contribution of this study are:

1. Developed an Automatically Constructed Dataset (ACD) using rule-based approach for Question Generation.
2. Proposed novel framework for anaphora resolution and word chunking.
3. Fine-tuned the multilingual transformer for Urdu language with different embedding, with and without proposed algorithms of anaphora resolution and word chunking.
4. Proposed an algorithm to rank generated question using semantic text representation.
5. Evaluated and compared the result for question generation obtained from each model using automatic and human evaluation.

2 Related Work

Heilman's (Heilman and Smith, 2009, 2010) research established a foundational rule-based framework for Automatic Question Generation (AQG) in English, which has influenced subsequent studies using dependency parsing, POS, and NER tagging, and semantic role labeling with rule-based systems (Khullar et al., 2018; Azevedo et al., 2020; Flor and Riordan, 2018; Dhole and Manning, 2020). Over the past decade, AQG has incorporated machine learning and deep learning techniques, such as RNNs for sequence transduction (Du et al., 2017), reinforcement learning with graph-to-sequence models (Chen et al., 2019), and transformers for faster training (Kriangchaivech and Wangperawong, 2019; Goyal et al., 2024). Notable models include the use of T5 for inferential questions (Ghanem et al., 2022), pre-trained BART on an inquisitive dataset (Gao et al., 2022), and a neural question generator trained on diverse datasets (Murakhovs'ka et al., 2022). Additionally, EQG-RACE integrates pre-trained BERT and ELMo embeddings with an Answer-guided Graph Convolution Network (Jia et al., 2021), while another framework uses pre-trained embeddings on BERT and EMLo, trained with GPT and GPT2 (Yuan et al., 2021). A model combining BiLSTM with soft attention and layers for encoding and decoding has also been proposed (Bi et al., 2021). Large language models like GPT-3 and ChatGPT have further advanced AQG with their extensive pre-trained knowledge and sophisticated language understanding (Lee et al., 2023). These language models also support multilingual question generation, but their evaluation in low-resource languages has yet to be explored.

Furthermore, there is a growing interest in extending these capabilities to low-resource languages like Arabic, Hindi, Bengali, which often lack the extensive labelled datasets and advance language processing tools (Kazi et al., 2023).

For the Hindi language, (Anuranjana et al., 2019) proposed a rule-based AQG system utilizing POS tagging, NER tagging, and dependency parsing, enhanced with linguistic rules and IndoWordNet ontology to generate questions. Surface-level and syntactic filters were applied to improve question quality, but these filters sometimes removed important questions containing pronouns for example:
Passage: Nelson Mandela was the first president of South Africa. He was born on 18 July 1918
Question: When was he born?

This question was removed by the filter but "When Nelson Mandela was born?" is the important question. In our proposed frame work, we proposed algorithm for anaphora resolution that replace pronoun with appropriate noun instead of just removing the question with that pronoun. On other hand, (Kumar et al., 2019) proposed a cross-lingual AQG system (CLQG) for Hindi and Chinese, using a shared encoder-decoder architecture trained in two phases: unsupervised training with denoising, autoencoding, and back-translation, followed by supervised training with sequence-to-sequence modeling achieving maximum score of 20.242, 29.143, and 40.643 for BLEU-4, METEOR and ROUGE-L. (Wang et al., 2021) proposed a multilingual language model for automatic question generation in five languages, including Hindi and Chinese, utilizing deep learning models such as Transformer and Multi-BERT achieving highest scores of 35.19 for BLEU-4, 36.25 for METEOR, and 51.23 for ROUGE for Hindi language.

For Arabic, Arabic Question Generator(AQG) claim to be first automatic question generation system since earlier proposed system were semi-automatic (Bousmaha et al., 2020). It combines rule based approach with the semantic role labelling of PropBank (SRL) to generate questions automatically from Arabic text. (Alhashedi et al., 2024) proposed arabic automatic question generation using transformers and scores achieved were 19.12 for BLEU-4, 23.00 for METEOR, and 51.99 for ROUGE-L.

For Bengali language, (Fahad et al., 2024) trained three different answer agnostic transformer model BanglaT5, mT5- base, BanglaGPT2 with different combination of decoding algorithm to generate questions automatically. The scores achieved by their best performing models were 11.42 for BLEU-4, 21.79 for METEOR, and 35.74 for ROUGE-L. On other hand, (Ruma et al., 2023) trained BanglaT5, Mt5-small, Mt5-base transformer model along with the answer for automatic question generation and best model achieved 36.60 Bleu-4, 48.98 METEOR, and 63.38 ROUGE-L scores.

Notably, little to no significant work has been done for AQG in the Urdu language, highlighting a gap in this area of research. To our best knowledge, there is no publication for Urdu language AQG till the writing of this research paper. Hence, to address this gap, the study proposed a hybrid automatic question generation framework that incorporates a rule-based approach with a deep learning model customized for the Urdu language. However, the the framework could be applicable to any language by customizing the rules specific to the language and training the modules of the framework on a corpus specific to that language.

3 Methodology

The proposed framework integrates a rule-based approach with a transformer model and comprises the following seven stages, as seen in Figure 1:

1. Pre-processing
2. Tagging, which includes POS, NER, and dependency parsing
3. Algorithm for Word Chunking
4. Algorithm for Anaphora Resolution
5. Development of Automatically Constructed Dataset (ACD) using rule based approach
6. Fine-tuning multilingual T5 (mT5) model with combination of different embedding and proposed algorithm on ACD and UQuAD 1.0 (Kazi and Khoja, 2021).
7. Ranking of generated questions

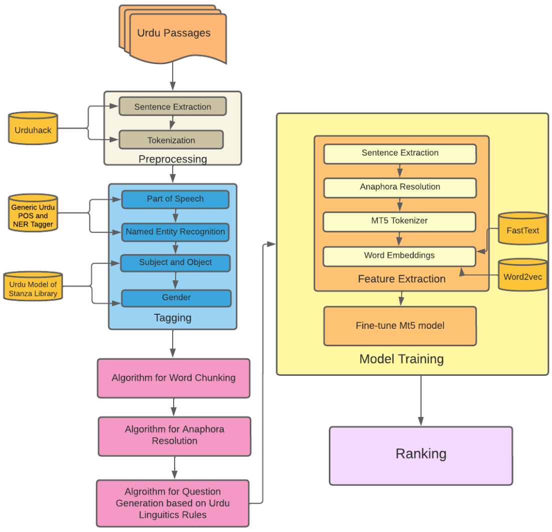


Figure 1: Proposed Framework

3.1 Pre-processing and tagging

Sentence extraction, word tokenization and normalization for each passage are performed using UrduHack (ALi, 2020). The POS are tagged using (Nasim et al., 2020) having F1 score of 96%, NER are tagged using (Kanwal et al., 2019) having F1 score of 77% and the dependency tree is extracted using the Urdu Model of Stanza Library (ALi, 2020).

3.2 Algorithm for Word Chunking

Even after using a word tokenizer, the single word can be treated as multiple words, for example, in the sentence, محمد علی جناح بانی پاکستان ہیں۔ (Muhammad Ali Jinnah is the founder of Pakistan), the word محمد علی جناح (Muhammad Ali Jinnah) is treated as three different words instead one single word. To solve this issue, an algorithm for Urdu noun chunking is developed. The algorithm identifies the group of noun and adjectives that goes together and chucks them together by removing space and putting "-" to form a meaningful word. The steps for the noun chunking algorithm are as follows:

- If a part of speech is followed by the same part of speech, for example, a noun is followed by the same type of noun, they are chunked together. In the sentence above, محمد,علی and جناح are proper nouns (PN), so they are chunked together. However, بانی is a common noun (NN), so it is not chunked together. The chunked word will be محمد-علی-جناح.

- However, chunking the same type of noun is not enough and can result in incorrect chunking, for example, in the sentence محمد علی جناح کراچی میں پیدا ہوئے۔ (Muhammad Ali Jinnah was born in Karachi), محمد-علی-جناح-کراچی (Muhammad Ali Jinnah Karachi) will be chunked together. This is an example of incorrect chunking as کراچی (Karachi) should not be chunked with محمد-علی-جناح (Muhammad Ali Jinnah). To solve this issue, NER tagging is also taken into consideration. The same type of noun is only chunked together if it has the same named entity tag. Since محمد,علیand جناح are tagged as a person, the words will be chunked together, but کراچی will not be chunked with it since its NE tag is location.

- If a noun chunk is preceded by an adjective, the adjective is also chunked with the noun, for example, in the sentence ثناء نے کالی ٹوپی پہن رکھی ہے۔ (Sana is wearing a black cap), the noun ٹوپی(cap) is chuck with the adjective کالی (black).

- If the adjective and noun chunk is preceded by the adverb, the adverb is also chunk with the noun, for example, in the sentence, یہ ایک بہت پرانی عمارت ہے (This is a very old building), the adverb بہت (very), the adjective پرانی (old) and the noun عمارت (building) are chunk together.

- If there is a conjunction between the same type of noun and the named entity tagged, they are chunked together. For example, جناح نے بمبے یونیورسٹی اور لنکن ان لندن سے تعلیم حاصل کی۔ (Jinnah Studied from Bombay university and Lincoln Inn London.), the terms بمبے-یونیورسٹی-اور-لنکن-ان-لندن are chunk together.

In addition to aiding in rule formation, the word chunking algorithm can be utilized to train deep and large language models by using word chunks as answers, particularly in datasets where answers are not available and only passages are provided. The experiments conducted in Section 4 indicate that passing word chunks as answers to mT5 performs better than using answer agnostic transformer.

3.3 Algorithm for Anaphora Resolution

The sentences are extracted from the input passages. Each individual sentence might contain a pronoun referring to a noun in previous sentences. Separating each sentence might result in the pronoun becoming ambiguous. To avoid this ambiguity, a noun and pronoun agreement algorithm has been developed. The algorithm replaces subject pronouns with the corresponding noun from the previous sentence. If the previous sentence does not contain a subject noun, the algorithm continues to look back through earlier sentences until it finds a subject noun, replacing all subject pronouns accordingly. Similarly, object pronouns are replaced with the corresponding object noun from previous sentences, with the last occurring object noun being used to replace the object pronoun in the current sentence.

3.4 Development of ACD

Urdu linguistic rules are employed for the generation of questions from provided passages. In the process, 250 passages were subjected to rule-based Automatic Question Generation (AQG). These passages were sourced from diverse outlets such as Urdu Wikipedia, Urdu stories from Urdu Point, and online Urdu comprehension materials. The resultant dataset, named ACD (Automatically Constructed Dataset), comprises 250 passages with 4497 question-answer pairs. The dataset is , hosted on a private GitHub repository, and it can be requested from by emailing corresponding author at mrkhowaja@iba.edu.pk.The types of questions generated, and the rules applied to generate those questions are as follows:

1. Who Questions (کون/کس)
   کون/کس question words are used depending upon whether the noun is oblique or normative. Oblique nouns are those noun which are followed by preposition whereas normative noun are not followed by preposition. For oblique noun, noun chunk is replaced by کس, for example, چوہدری رحمت علی نے پاکستان کانام رکھا (Chaudhry Rehmat Ali coined the name of Pakistan.) becomes کس نے پاکستان کانام رکھا؟. For normative noun, noun chunk is replaced کون, for example, جارج واشنگٹن امریکہ کے پہلے صدر تھے (George Washington was the first president of the United States) becomes کون امریکہ کے پہلے صدر تھے؟

2. What Questions (کیا)
   If the word chunk is the object noun, it is replaced by کیا, for example, in sentence بچہ رنگین تصویر بنارہاہے۔ (The child is drawing a colorful picture.) the رنگین تصویر (colorful picture) is replaced by کیا

3. Where Questions (کہاں)
   If the chunked noun is a location or organization, it is replaced by کہاں. For oblique noun, only chunk noun is removed for example, جناح نے بمبے یونیورسٹی اور لنکن ان لندن سے تعلیم حاصل کی (Jinnah studied at Bombay University and Lincoln's Inn, London.) becomes جناح نے کہاں سے تعلیم حاصل کی؟ (Where did Jinnah studied from?). For normative noun, chunk noun along with its preposition is removed, for example سارہ کراچی میں رہتی ہے (Sarah lives in Karachi) becomes سارہ کہاں رہتی ہے؟ (Where Sarah lives?)

4. When Questions (کب)
   Dates or times found by the NER tagger are replaced by کب along with their prepositions. For example: محمد علی جناح 25 دسمبر 1876 کو پیدا ہوئے (Muhammad Ali Jinnah was born on 25 December, 1876.) becomes محمد علی جناح کب پیدا ہوئے؟ (When was Muhammad Ali Jinnah born?)

5. How Many Questions (کیتنی/کیتنے)
   In Urdu, the choice between کیتنی/کیتنے for "How many" questions and کتنا for "How much" questions depends on whether the noun is countable or uncountable. For instance, in the question کتنا وقت لگے گا؟ (How much time will it take?), وقت (time) being uncountable uses کتنا (How much). However, Urdu lacks a noun tagger to distinguish countable and uncountable nouns, limiting the study to "How many" questions. When a cardinal value is identified, it is replaced with either کیتنی or کیتنے based on the gender of the dependent noun. For example, in the sentence کلاس میں 10 لڑکیاں ہیں۔ (There are 10 girls in the class), لڑکیاں (girls) being feminine, is replaced by کتنی, forming the question کلاس میں کتنی لڑکیاں ہیں؟ (How many girls are there in the class?). Similarly, in کلاس میں 10 لڑکے ہیں۔ (There are 10 boys in the class), لڑکے being masculine is replaced by کیتنے, forming کلاس میں کتنے لڑکے ہیں؟ (How many boys are there in the class?). When both masculine and feminine nouns are involved, کیتنے is used, as in کلاس میں 10 لڑکے اور لڑکیاں ہیں۔ (There are 10 boys and girls in the class), resulting in کلاس میں کتنے لڑکے اور لڑکیاں ہیں؟ (How many boys and girls are there in the class?). For oblique nouns like in ثنااس فرم میں 10 سال سے کام کررہی ہے۔ (Sana has been working in this firm for 10 years), سال (years), followed by the preposition سے, uses کیتنے, resulting in ثنااس فرم میں کیتنے سال سے کام کررہی ہے؟ (How many years has Sana been working in this firm?).

6. Why Questions (کیوں)
   The کیوں is used to ask why Question. When a sentence contains کیونکہ (because), the whole chunk after کیونکہ along with کیونکہ is removed and the word کیوں is placed at the beginning of the sentence. For example, in the sentence محمد علی جناح نے کانگریس چھوڑدی کیونکہ سیاسی اختلاف تھا (Muhammad Ali Jinnah left congress because of political difference.), the whole chunk کیونکہ سیاسی اختلاف تھا is removed and کیوں is placed at the beginning of the sentence. The question formed will be کیوں محمد علی جناح نے کانگریس چھوڑدی؟

The question generated were evaluated by human evaluator on the basics of syntax, semantics and relevance on 10-likert scale. Question with average accuracy of less than 50% on any of these metrics were removed. The final set of data of 250 passages and 4497 questions as seen in Table 1 and distribution of types of questions can be seen in Figure **??**.

Metrics	Value
Total Passages	250
Total Questions	4497
Average Length of Question	14
Average Length of Paragraph	9 Sentences
Types of Questions	6
Maximum Length of Question	31
Number of Who Questions	989
Number of What Questions	855
Number of When Questions	900
Number of How many Questions	719
Number of Why Questions	90

Table 1: ACD Statistics

3.5 Automatic Question Generation using Multilingual Transformers

In this framework, a text-to-text transformer architecture is proposed for automatic question generation, favoring transformers over models like RNNs and LSTMs due to their efficiency in capturing long-range dependencies and faster training speed. This efficiency is attributed to the self-attention mechanism in transformers, allowing each token to attend to all others in the sequence. We employed Multilingual T5 (MT5) as it integrates both encoder and decoder models trained on various languages. This architecture allows us to encode context (passage) and answer and decode generated question as the output. The sample output is as follows:

Context:

جناح نے 1913 سے 14 اگست 1947 کو پاکستان کے قیام تک آل انڈیا مسلم لیگ کے رہنما کے طور پر خدمات انجام دیں، اور پھر اپنی موت تک پاکستان کے پہلے گورنر جنرل کے ڈومینین کے طور پر خدمات انجام دیں۔

Answer:

جناح

Questions:

کون 1913 سے 1947 تک آل انڈیا مسلم لیگ کے رہنما تھے؟
کون 1913 سے 1947 تک آل انڈیا مسلم لیگ کے سربراہ تھے؟
کس نے 1913 سے 1947 تک آل انڈیا مسلم لیگ کی قیادت کی؟

To enhance context utilization and address computational constraints, the MT5 model is trained at the sentence level rather than the passage level. Sentences are extracted and anaphora resolution is applied to resolve ambiguous references, followed by tokenization using sentence piece. To capture similar words, Word2Vec and FastText word embeddings are employed. The encoder maps each word in a sentence to a dimensional vector, and sentences are padded to a maximum length of 512 for uniformity. The self-attention mechanism in the encoder captures dependencies between words in both local and global contexts. Input and target sequences are formatted and passed to the transformer for output generation. The decoder generates the output sequence using self-attention, with beam search employed to explore multiple candidate sequences simultaneously, enhancing the likelihood of capturing important and diverse questions. For example, consider the Urdu sentence, ہمالیہ میں واقع ماؤنٹ ایورسٹ دنیا کا سب سے اونچا پہاڑ ہے۔ (Mount Everest in the Himalayas is the highest mountain in the world.) with answer ہمالیہ (Himalayas). The greedy search made the following question دنیا کا سب سے اونچا پہاڑ کس میں واقع ہے؟ (Which is the highest mountain in the world located?) but the top questions using beam search were ماؤنٹ ایورسٹ کہاں واقع ہے؟ (Where is Mount Everest Located?), دنیا کا سب سے اونچا پہاڑ کہاں واقع ہے؟ (Where is the highest mountain in the world located?) and دنیا کا سب سے اونچا پہاڑ کہاں پایا جا سکتا ہے؟.(Where can the highest mountain in the world be found?) The snapshot of the probability calculated using beam search can be seen in Figure 2.

3.6 Ranking of Generated Questions

Multiple questions can be generated from a single sentence which results in an over generation of questions. To resolve this issue, the questions are ranked, and the top 10 questions are selected. Currently, the ranking algorithm is determined by their similarity to the original sentence. Sentence embeddings for each sentence in the paragraph are calculated using SBert multilingual. Subsequently, the embeddings of each sentence in the passage are averaged to create the passage's overall embedding. Likewise, sentence embeddings for the questions are computed. After which, cosine similarity is employed to measure the similarity between the passage and each question, and the top 10 questions with the highest similarity scores are selected as seen in Figure 3.

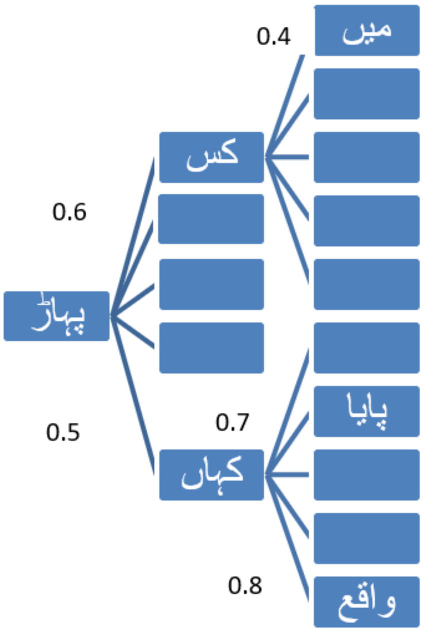


Figure 2: Probability Distribution of Beam Search Candidates

4 Experiment

The study conducted four different sets of experiments on the ACD and UQuAD 1.0 (Kazi and Khoja, 2021) datasets , utilizing pre-trained embeddings, specifically Urdu word embeddings (Haider, 2018) and FastText (Grave et al., 2018), integrated with a deep learning model both with and without anaphora resolution and a ranking algorithm. Additionally, various hyper-parameters such as learning rate, number of epochs, and batch size were systematically adjusted to optimize the model's performance. The model performed best with the hyper parameters shown in Table 2, considering the computational limitations.

Hyperparameter	**Value**
Epochs	3
Optimizer	Adam
Batch Size	12
Learning Rate	1e-5
Number of Beams	5
Number of Sequence	3

Table 2: Optimal Hyper parameters for Model Performance

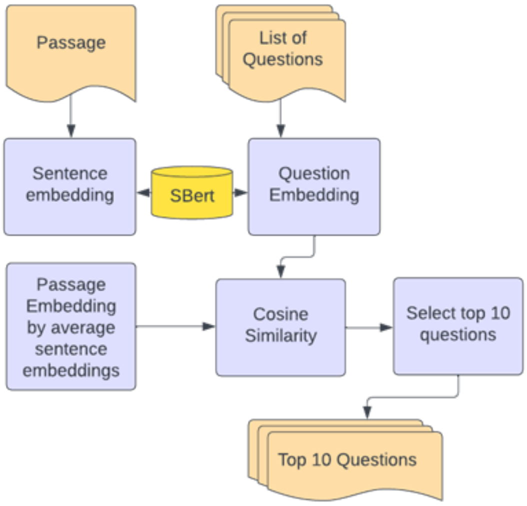


Figure 3: Proposed Ranking Algorithms for Generated Questions

- Experiment 1: Fine-tuned Mt5 on ACD with Urdu word embeddings (Haider, 2018) and FastText (Grave et al., 2018).
- Experiment 2: Fine-tuned Mt5 on ACD along with two specified embeddings and anaphora resolution named Sawaal.
- Experiment 3: Fine-tuned Sawaal along with ranking algorithm.
- Experiment 4: Fine-tuned best performing model i.e. MT5 along with anaphora resolution and ranking algorithm on FastText embeddings on the following dataset and encoding:
    1. Answer Aware MT5 trained on ACD.
    2. Answer Agnostic MT5 trained on ACD.
    3. Word Chunks encoded as answer for MT5 trained on ACD.
    4. Answer Aware MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).
    5. Answer Agnostic MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).
    6. Word Chunks encoded as answer for MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).

The experiments were designed to analyze the accuracy of the proposed anaphora resolution and ranking algorithms. By comparing these experiments, the study aimed to gain insights into the effectiveness of different word

Model	BLEU-4	METEOR	ROUGE-L
MT5 + Urdu embeddings	17.73	20.97	39.89
Sawaal + Fast Text embeddings	21.36	35.23	52.38
Sawaal + Urdu embeddings	22.14	34.02	51.65
Sawaal + Fast Text embeddings + Ranking	24.78	37.07	54.99
Sawaal + Urdu embeddings + Ranking	23.32	36.47	53.66

Table 3: Evaluation Metrics for Question Generation Models

embedding techniques and the proposed algorithms when combined with large learning models like MT5. Specifically, the comparison between experiment 1 and experiment 2 highlights the impact of the anaphora resolution algorithm on improving model accuracy. The comparison between experiment 2 and experiment 3 evaluates the effectiveness of the ranking algorithm. Finally, experiment 4 assesses the model's adaptability and generalization capabilities when trained on different datasets, while also evaluating the efficiency of using word chunks for answer encoding within the MT5 framework.

5 Evaluation

The questions produced by the framework undergo evaluation against the UQuAD 1.0 test dataset to compute metrics such as F-scores for METEOR (Banerjee and Lavie, 2005), BLEU-4(Papineni et al., 2002), and ROUGE-L (Lin, 2004). Table 3 presents the scores achieved by each model in the experiment 1-3, utilizing the following encoding format:

input = ”context: %s answer: %s </s>” %
target = ”question: %s </s>” %

While Table 4 and Figure 4 presents the scores of fine-tuning the T5 model on different datasets and combination of various input encoding method.

Combinations	Datasets	Encoding
1	UQuAD 1.0	(p,a)
2	UQuAD 1.0	(p)
3	UQuAD 1.0	(p,wc)
4	ACD	(p,a)
5	ACD	(p)
6	ACD	(p,wc)

Table 4: Encoding Combinations for Datasets. p stands for paragraph, a stands for answer and wc stands for chunks

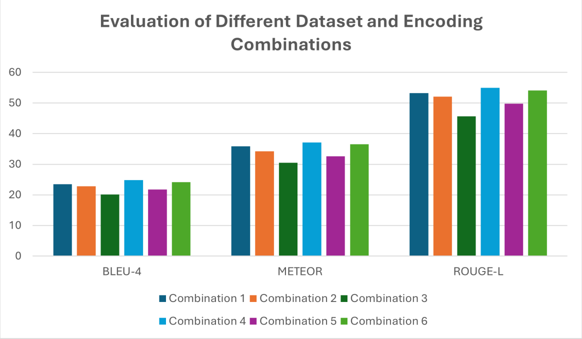


Figure 4: Evaluation of Different Dataset and Encoding Combinations

The study also evaluated final set of generated question from 10 human expert who evaluated the question based on syntax, relevance and semantics of the question on 10-likert scale. The average score for syntax, semantics and relevance were as 8.4, 8.2 and 7.7 respectively.

Limitations

While the proposed framework is able to generate semantically, syntactically and relevant questions from the passage it also have few drawbacks. Firstly, both datasets used for model training contain errors. UQuAD 1.0 contains translation errors, while ACD suffers from errors due to rule exceptions, insufficient language processing tools, incorrect tagging. Secondly, in the studies only pre-trained word embeddings are used. Future research aims to train conceptual embedding such as mBERT and SBERT for urdu language to enhance semantic understanding and context in natural language processing tasks. Thirdly, current system limitations include treating all sentences equally in importance and relying solely on similarity measures for question ranking, without considering whether the answer is present in the passage, which are aspects intended for future implementation.

Ethics Statement

This study adheres to the highest ethical standards in research. All data used, including text passages and question-answer pairs, were sourced from publicly available materials and databases, ensuring that no private or sensitive information was used without explicit consent. Additionally, all sources of data have been properly cited, and the use of copyrighted materials complies with relevant laws and guidelines.

References

Saleh Saleh Alhashedi, Norhaida Mohd Suaib, and Aryati Bakri. 2024. Arabic automatic question generation using transformer model. In *AIP Conference Proceedings*, volume 2991. AIP Publishing.

Ikram ALi. 2020. Urduhack: A python library for urdu language processing.

Kaveri Anuranjana, Vijjini Anvesh Rao, and Radhika Mamidi. 2019. Hindi question generation using dependency structures.

Pedro Azevedo, Bernardo Leite, Henrique Lopes Cardoso, Daniel Castro Silva, and Luís Paulo Reis. 2020. Exploring nlp and information extraction to jointly address question generation and answering. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 396–407. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Lizhen Qu, Shirong Shen, Guilin Qi, Lu Pan, and Yinlin Jiang. 2021. Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model. *arXiv preprint arXiv:2110.06560*.

Kheira Zineb Bousmaha, Nour H Chergui, Mahfoud Sid Ali Mbarek, and Lamia Hadrich Belguith. 2020. Aqg: Arabic question generator. *Rev. d'Intelligence Artif.*, 34(6):721–729.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. *arXiv preprint arXiv:1908.04942*.

Ali Daud, Wahab Khan, and Dunren Che. 2017. Urdu language processing: a survey. *Artificial Intelligence Review*, 47:279–311.

Kaustubh D Dhole and Christopher D Manning. 2020. Syn-qg: Syntactic and shallow semantic rules for question generation.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension.

Abdur Rahman Fahad, Nazme Al Nahian, Md Ahanaf Islam, and Rashedur M Rahman. 2024. Answer agnostic question generation in bangla language. *International Journal of Networked and Distributed Computing*, pages 1–26.

Michael Flor and Brian Riordan. 2018. A semantic role-based approach to open-domain automatic question generation. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 254–263.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. 2022. "what makes a question inquisitive?" a study on type-controlled inquisitive question generation. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 240–257, Seattle, Washington. Association for Computational Linguistics.

Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. Question generation for reading comprehension assessment by modeling how and what to ask. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2131–2146, Dublin, Ireland. Association for Computational Linguistics.

Rupali Goyal, Parteek Kumar, and VP Singh. 2024. Automated question and answer generation from texts using text-to-text transformers. *Arabian Journal for Science and Engineering*, 49(3):3027–3041.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Samar Haider. 2018. Urdu word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2021. Eqg-race: Examination-type question generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13143–13151.

Safia Kanwal, Kamran Malik, Khurram Shahzad, Faisal Aslam, and Zubair Nawaz. 2019. Urdu named entity recognition: Corpus generation and deep learning applications. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–13.

Samreen Kazi and Shakeel Khoja. 2021. Uquad1. 0: Development of an urdu question answering training data for machine reading comprehension.

Samreen Kazi, Shakeel Khoja, and Ali Daud. 2023. A survey of deep learning techniques for machine reading comprehension. *Artificial Intelligence Review*, 56(Suppl 2):2509–2569.

Payal Khullar, Konigari Rachna, Mukul Hase, and Manish Shrivastava. 2018. Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*, pages 153–158.

Kettip Kriangchaivech and Artit Wangperawong. 2019. Question generation by transformers.

Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-lingual training for automatic question generation.

Philippe Laban, Chien-Sheng Wu, Lidiya Murakhovs'ka, Wenhao Liu, and Caiming Xiong. 2022. Quiz design task: Helping teachers create quizzes with automated question generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 102–111, Seattle, United States. Association for Computational Linguistics.

Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 2023. Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education. *Education and Information Technologies*, pages 1–33.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lidiya Murakhovs'ka, Chien-Sheng Wu, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. 2022. MixQG: Neural question generation with mixed answer types. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1486–1497, Seattle, United States. Association for Computational Linguistics.

Zarmeen Nasim, Shaukat Abidi, and Sajjad Haider. 2020. Modeling pos tagging for the urdu language. In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–6. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jannatul Ferdous Ruma, Tasmiah Tahsin Mayeesha, and Rashedur M Rahman. 2023. Transformer based answer-aware bengali question generation. *International Journal of Cognitive Computing in Engineering*, 4:314–326.

Bingning Wang, Ting Yao, Weipeng Chen, Jingfang Xu, and Xiaochuan Wang. 2021. Multilingual question generation with language agnostic language model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2262–2272.

Wei Yuan, Tieke He, and Xinyu Dai. 2021. Improving neural question generation using deep linguistic representation. In *Proceedings of the Web Conference 2021*, pages 3489–3500.

Thonburian Whisper: Robust Fine-tuned and Distilled Whisper for Thai

Zaw Htet Aung
Department of Biomedical Engineering
Faculty of Engineering
Mahidol University
Thailand
`zawhtetaung.zaw@student.mahidol.ac.th`

Thanachot Thavornmongkol
Looloo Technology
Thailand
`kimmchi.thanachot@loolootech.com`

Atirut Boribalburephan
Looloo Technology
Thailand
`atirut.name@loolootech.com`

Vittavas Tangsriworakan
Looloo Technology
Thailand
`vittavas.tan@student.mahidol.edu`

Knot Pipatsrisawat
Looloo Technology
Thailand
`knot@loolootech.com`

Titipat Achakulvisut
Department of Biomedical Engineering
Faculty of Engineering
Mahidol University
Thailand
`titipat.ach@mahidol.edu`

Abstract

Despite extensive pre-training on a large audio corpus, the Whisper and Distil-Whisper models exhibit considerable challenges in handling Thai speech. This paper presents an approach to enhance pre-trained vanilla Whisper models for Thai automatic speech recognition (ASR). The process involves combining audio datasets, applying audio augmentations during training, and incorporating an audio segmentation strategy. In addition, we show that distilling whisper models can be achieved with less than 1,500 hours of audio while preserving accuracy of student models. The improved models achieve a word error rate (WER) of 11.01%, 6.62%, 5.49%, 11.23%, 7.57% for the small, medium, large, distill-small, and distill-medium Whisper models on Commonvoice 13 dataset. Our models establish as a fine-tuned baseline Whisper ASR for Thai. Furthermore, we demonstrate accuracy of our models with out-of-distribution (OOD) financial datasets while maintaining robustness under environmental noise. The code and pretrained models are available at `https://github.com/biodatlab/thonburian-whisper/`.

1 Introduction

Automatic Speech Recognition (ASR) converts spoken language into text, which enables several applications such as audio transcription and conversational analysis. Contemporary deep learning-based systems such as Wav2Vec2 (Baevski et al., 2020), Conformer (Gulati et al., 2020), Massively Multilingual Speech (MMS) (Communication et al., 2023), Whisper (Radford et al., 2023), and Seamless M4T (Communication et al., 2023) have demonstrated impressive capabilities in the conversion of spoken languages into text in both English and multilingual audios. However, their performance diminishes when applied to languages with limited audio resources (Bansal et al., 2019). Moreover, adapting these models to accurately transcribe audio in language-specific and specialized domains remains challenging.

Previous efforts to improve Thai ASR models include Wav2Vec2-XLSR (Baevski et al., 2020), Thai Wav2Vec 2.0 (Phatthiyaphaibun et al., 2022) , MMS and Seamless M4T (Communication et al., 2023), which scaled up the Wav2Vec2 architecture to over 1,000 languages. Even though these models perform well in English speech, their performance limitations have been observed in bilingual datasets (Abushariah et al., 2023) and out-of-domain language specific datasets (Jain et al., 2023). This is common in Thai financial audio reports and conference calls, in which most financial terms and company names are dominated by non-native accented English. Inaccurate recognition not only increases the word error rate (WER) but also degrades downstream tasks such as information extraction. End-to-end transformer-based architectures such as OpenAI's Whisper (Radford et al., 2023) have shown promising results in ASR tasks. Whisper is extensively pretrained on a large multilingual audio corpus of 680,000h, potentially making it a robust and reliable ASR system for Thai speech. This presents an opportunity to combine the strength of transformers and a larger and more diverse datasets to improve the performance of ASR models for Thai.

In this study, we enhance the existing Whisper ASR models by creating a collection of open Whisper models specifically designed for Thai. We combine multiple audio corpora from various sources for fine-tuning. Our main objective is to build a diverse corpus that captures the range of speech nuances, dialects, and accents in Thai language. To enhance the robustness of our models, temporal and spectral augmentations were introduced during fine-tuning. We experimented with models trained using these enhancements to understand their impact on improving model performance and resilience against varying quality and background noise. Balancing model accuracy with computational efficiency is an important consideration especially for environments with limited resources. Previous works (Gandhi et al., 2023) show that it is possible to compress the Whisper models through knowledge distillation. However, a substantial amount of training data is needed for the distilled models to achieve comparable performance to their counterparts. Our work showed that it is possible to achieve successful model compression for Whisper models using a fraction of training data used in (Gandhi et al., 2023). We show significant reductions in word error rates (WER) in all model sizes compared to vanilla Whisper and other ASR models for Thai. Finally, we demonstrate our model's adaptability on OOD financial data. We release the code and pretrained models which can be used as baselines for Thai Whisper ASR.

1.1 Related Works

Availability of transformer-based multilingual ASR models pretrained on massive datasets marks a milestone in the field of low-moderate resource ASR. Yet, few works have addressed the challenges associated with building a robust ASR for Thai. Naowarat et al. introduced contextualized connectionist temporal classification (CCTC) loss to address spelling inconsistencies in code switching Thai ASR. The contextual prediction capabilities inherent in transformer architectures such as those seen in Whisper models align with the objectives of the CCTC loss. The study focusing on ASR technology for Thai dialects (Suwanbandit et al., 2023) highlighted the importance of understanding tonal variations and employing targeted learning approaches to enhance Thai ASR accuracy. Due to the diverse language landscape of Thailand, models capable of handling dialectical differences are needed. Another advancement is the introduction of fine-tuned Wav2Vec2 models for Thai (Phatthiyaphaibun et al., 2022). Here, they utilized a self-supervised pretrained Wav2Vec2 model and fine-tuned on the Commonvoice dataset. However, the total training data was only 128 hours. A more comprehensive evaluation and pretrained models were needed to understand the model's capabilities. Recent development of transformer-based models such as Whisper (Radford et al., 2023) and Distil-Whisper (Gandhi et al., 2023) have shown to effectively capture robustness in multiple languages. By extending the scope of training to languages with limited resources, such as Thai, we can acquire critical insights into the process of fine-tuning these models. This effort will contribute to the accessibility of Thai ASR within the research community.

2 Materials and Methods

2.1 Datasets

2.1.1 Pretraining datasets

We aim to improve Whisper models to robustly transcribe Thai audios. The first stage is to collect data sets to pretrain the Whisper models. We combine multiple primary data sources for pretraining from publicly available speech and internet audio datasets, including Thai CommonVoice 13 (CMV13) (51.41h) (Ardila et al., 2020), Google Fleurs (8.49h) (Conneau et al., 2023), Gowajee (15h) (Chuangsuwanich et al., 2020) and Thai Elderly Speech (26.56h)[1], and Thai Central Dialect corpus (683.9h) (Naowarat et al., 2021). To make the model generalize to most domains, we scrape audio from various sources on the Internet, first listing 250 generic Thai keywords and exploring their associated queries or topics using Google Trends[2]. We then used the associated queries to search for audios over the Internet and acquired 5,100 uncleaned captioned audios. To clean the captioned audio, the audios are selected if they are (i) pub-

---

[1] https://github.com/VISAI-DATAWOW/Thai-Elderly-Speech-dataset/releases/tag/v1.0.0

[2] https://trends.google.com/trends/

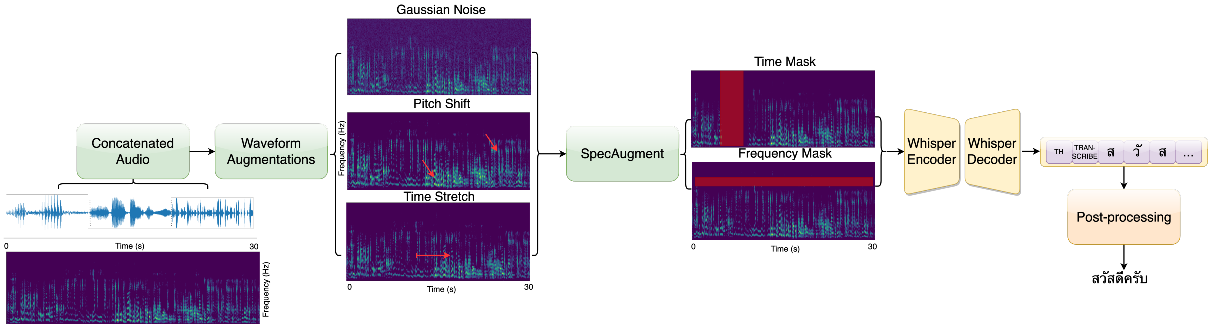


Figure 1: **Schematic of model pretraining**: The Whisper model is fine-tuned on a collection of more than 1.3k hours of Thai audios with additional augmentations including audio clip concatenation, waveform augmentation (Gaussian noise, time stretch, and pitch shift), and spectral augmentations.

licly accessible, (ii) in Thai, (iii) available with Thai subtitles, and (iv) not categorized as music, resulting in a total of 631.89 hours of additional audios. Combining these datasets results in a total of 1,316.76 hours.

2.1.2 OOD Financial audio dataset

We have assembled a specialized dataset tailored for the financial domain to see adaptability and usage of the models in domain-specific use cases. The rationale behind this lies in the presence of distinctive terminology within the financial sector, which is not typically encountered in general-purpose databases. In this effort, we collected around 18 hours of financial reports from earnings call videos, "Oppday"[3]. These records cover quarterly reports from various companies from 2020 to 2021. We used VAD (Team, 2021) to segment each audio file into short snippets ranging approximately from 2 to 4 seconds, resulting in 30,185 audio snippets. We annotate these snippets using the web-based tool 'Audino' (Grover et al., 2021). The audio samples are divided into 28,568 (96.64%, 17.69h) and 1,617 (5.36%, 1.08h) samples of training and testing, respectively. We use the OOD data to evaluate zero-shot generalization and fine-tune our models to see their adaptability compared to other Whisper models.

2.2 Thonburian Whisper Pretraining

The performance of the fine-tuned Whisper for Thai depends on the size of the pre-training and the fine-tuning strategy. Here, we select small, medium, and large (v3) Whisper model sizes for fine-tuning for Thai. We fine-tuned the models using a straightforward approach without augmentation. In addition, we propose a set of augmentation techniques applied during training to make Whisper more robust for Thai audios, which include

- Concatenation of audio clips: The concatenation of short audio to reach the default Whisper input length (30s) improves the efficiency of the sample and training.
- Waveform augmentation: Raw waveform augmentations for collected audios are applied randomly. Techniques include Gaussian noise injection, temporal waveform dilation, and pitch shifting (Jordal et al., 2023).
- SpecAugment: We applied SpecAugment (Park et al., 2019) to mask the features of the spectrogram along the temporal and frequency axes. We set a probability of 0.3 for time masking and apply masking along 10 consecutive time steps. We applied frequency masking across 64 frequency bands with a probability of 0.1.

All models were trained for 10,000 iterations with and without proposed augmentation. We used a batch size of 16, using the deep-speed ZeRO optimizer (Rajbhandari et al., 2020). Pretrained models are evaluated and compared with the vanilla Whisper models, Thai Wav2Vec 2.0, and Seamless-M4T large models.

2.3 Model Distillation

We use the distillation technique proposed by (Gandhi et al., 2023) using layer-based compression of the Whisper decoder layers. Four maximally spaced decoder layers are copied from the teacher model to the student model, while the

---

[3] https://listed-company-presentation.setgroup.or.th/en

teacher's encoder layers are kept intact (Supplementary Table 4). During distillation, the prediction probabilities of the student model are trained to match those of the teacher model by minimizing the Kullback-Leibler (KL) divergence across the entire spectrum of possible tokens. In contrast to Gandhi et al. which used more than 21k hours of audio, including pseudo-labels, our approach utilizes a more modest distillation dataset of around 1,317 hours. Distillation is carried out in two steps: for the first 10,000 iterations, the optimal alignment between the encoder and decoder layers of the student model is achieved through the guidance of the teacher. Next, the student model is fine-tuned for another 10,000 iterations using the same dataset without relying on KL divergence loss. This approach of dual-step distillation and fine tuning allows the proposed distilled models to preserve accuracy despite utilizing significantly fewer hours of data.

2.4 Zero-shot Performance and Fine-tuning on OOD Financial Dataset

For vanilla Whisper (Radford et al., 2023) models, ThaiWav2Vec 2.0 (Phatthiyaphaibun et al., 2022) and Seamless M4T large (Communication et al., 2023), we evaluated their zero-shot performances on CMV13 test, FLEURS test and Thai Central dev datasets. Furthermore, we test all the models' zero-shot generalization on the OOD dataset and perform fine-tuning of both vanilla and Thonburian Whisper models to see their adaptability in the financial domain.

2.5 Model Robustness Under Environmental Noises

To evaluate the robustness of our pretrained models, we inject environmental disturbances sourced from the ESC-50 dataset (Piczak, 2015) into the FLEURS test set (Conneau et al., 2023). We used 2,000 environmental audio recordings that span 50 semantic categories, each lasting 5 seconds. We selected 40 longest-duration samples and adjusted the amplitude, using the noises according to the signal-to-noise ratio (SNR). Noise samples are duplicated or trimmed depending on the length of the audio to be inserted. This process is repeated across 9 SNR levels, ranging from - 20dB to 20dB with 5dB increments. This results in a corrupted test set that contains 2,000 corrupted audios for each SNR.

2.6 Model Evaluation

We perform naive text post-processing to normalize the output transcript, such as vowel corrections, tone mark orders, and extra white space removal. Evaluation is carried out by calculating the the word error rate (WER), the deletion error rate (DER), the substitution error rate (SER) and the insertion error rate (IER) with Thai word tokenizer, deepcut (Kittinaradorn et al., 2019). IER can be used to indicate the hallucination of the model, i.e., predicting repeated words. Other evaluation includes measurement of the latency in predicting short- and long-form audios (Supplementary Table 4).

3 Results and Discussion

3.1 Model Performance

We evaluate all models on the short-form audios without timestamp prediction on Common Voice 13, FLEURS, and Thai Central development datasets (Table 1). Thonburian Whisper have shown improved performance in all model sizes. They have shown less vulnerability to hallucinations as seen in the lower IERs. The small model gained the highest WER improvement after fine-tuning on the combined Thai dataset where the large Thonburian models obtained the lowest WERs on all our test sets. An interesting observation is that the augmented large model demonstrates a minor decline in performance on Commonvoice 13 and FLUER while slightly outperforming the non-augmented variant in Thai Central development dataset. The augmented models show a higher robustness after a noise corruption with SNR less than -5 dB (Figure 2).

Distilled Thonburian Whisper (S, M) with 1.3k hours of audios have shown comparable performance (less than 1 WER difference on CMV13 and Thai Central Dev) to the original model in all evaluated dataset (Table 1). They have 68.6% and 56.02% less parameters compared to the original S, M models. Hence, the distilled models achieve 1.26x and 3.89x speed up in short-form inference and 1.72x and 2.41x for long-form inference (Table 4). Therefore, the trade-off between accuracy and computational complexity may be justifiable in resource constrained scenarios.

Table 1: Evaluation Results on Different Datasets

Model	Params	CMV13-Test				Google Fleurs Test				Thai Central Dev			
	(M)	WER	IER	SER	DER	WER	IER	SER	DER	WER	IER	SER	DER
Vanilla (S)	242	38.8	8.6	26.7	3.5	43.0	8.6	30.5	3.9	61.5	5.0	41.3	15.1
Vanilla (M)	764	23.9	4.5	16.8	2.6	30.5	6.5	20.6	3.5	50.6	2.3	32.0	16.3
Vanilla (L)	1,543	12.8	2.1	9.1	1.5	14.7	3.2	9.4	2.0	37.9	1.9	22.6	13.3
Thonburian (S, A)	242	13.1	3.5	8.5	1.1	15.4	3.9	9.5	1.9	8.9	2.6	5.4	0.9
Thonburian (S)	242	11.0	2.2	7.7	1.1	14.1	3.3	8.8	2.0	8.7	2.6	5.1	1.0
Thonburian (M, A)	764	7.4	1.5	5.1	0.8	10.5	2.8	6.2	1.6	6.2	1.7	3.7	0.9
Thonburian (M)	764	6.6	1.0	4.8	0.8	10.2	2.8	5.9	1.5	6.8	2.4	3.7	**0.8**
Thonburian (L, A)	1,543	6.6	1.4	4.5	0.7	9.1	2.3	5.3	1.5	**5.4**	**1.3**	**3.2**	0.9
Thonburian (L)	1,543	**5.5**	**0.8**	**4.0**	**0.7**	**8.7**	**2.0**	**5.2**	**1.5**	6.0	1.8	3.3	0.9
Distilled Thonburian (S)	166	11.2	2.2	7.8	1.2	16.6	4.8	9.8	2.0	8.9	2.6	5.2	1.0
Distilled Thonburian (M)	428	7.6	1.2	5.5	0.9	12.5	3.4	7.3	1.8	6.5	1.6	3.9	1.0
Wav2Vec2 (L)	316	10.3	4.0	5.4	0.9	25.4	9.9	14.0	1.5	26.2	3.5	20.0	2.7
Seamless-M4T (L)	2,360	12.8	1.9	9.3	1.6	20.0	5.1	12.1	2.9	34.2	2.1	23.4	8.7

[1] S,M,L - Small, Medium, Large; A - Augmented

Table 2: Zero-Shot Performance on the OOD Financial Domain Test

Model	WER	IER	SER	DER
Vanilla (S)	72.7	26.6	31.8	14.4
Vanilla (M)	59.7	21.8	24.0	13.9
Vanilla (L)	25.2	3.3	12.4	9.5
Thonburian (S)	32.1	8.2	13.8	10.1
Thonburian (S, A)	33.2	10.9	14.0	8.3
Thonburian (M)	23.6	5.2	10.0	8.4
Thonburian (M, A)	25.4	8.2	10.0	7.2
Thonburian (L)	**18.7**	**2.5**	8.7	7.5
Thonburian (L, A)	19.7	2.6	**8.4**	8.7
Distilled Thonburian (S)	32.4	8.2	14.3	9.9
Distilled Thonburian (M)	27.5	5.4	11.5	10.6
Wav2Vec2 (L)	46.9	10.1	33.1	**3.7**
Seamless-M4T (L)	37.4	7.2	24.7	5.5

Table 3: OOD Fine-Tuning Results on Oppday Test Set

Model	WER	IER	SER	DER
Thonburian (S)	15.3	3.4	10.0	1.9
Thonburian (S, A)	15.2	3.3	9.9	1.9
Thonburian (M)	11.9	2.6	7.7	1.5
Thonburian (M, A)	**11.5**	**2.5**	**7.5**	**1.5**
Distilled Thonburian (S)	17.3	4.2	10.4	2.6
Distilled Thonburian (M)	13.3	3.0	8.6	1.7
Vanilla (S)	21.8	5.5	14.0	2.3
Vanilla (M)	14.7	3.3	9.6	1.7

3.2 OOD in financial domain and fine-tuning capability

Table 2 provides an analysis of how Thai ASR models perform when faced with OOD data without any additional fine tuning. Vanilla whisper models, especially small (72.7% WER) and medium (59.7% WER) ones, exhibit a significant struggle when dealing with audio samples from specific domains such as finance. Frequent code-switching between domain specific terms in English and Thai coupled with non-native accents makes it particularly challenging. In contrast, Thonburian Whisper models show remarkable improvements in performance compared to their vanilla counterparts. Models such as Wav2Vec2 (L) and Seamless-M4T (L) demonstrate higher WERs than Vanilla Whisper (L). In particular, their substitution error rates are much higher. This underscores the varying levels of success in zero-shot generalization across different model architectures. Table 3 shows the results on Oppday test set after fine-tuning on the domain specific data. All Thonburian models perform better than their vanilla Whisper counterparts. Interestingly, even the distilled models can adapt better to OOD data. This suggests that the proposed training

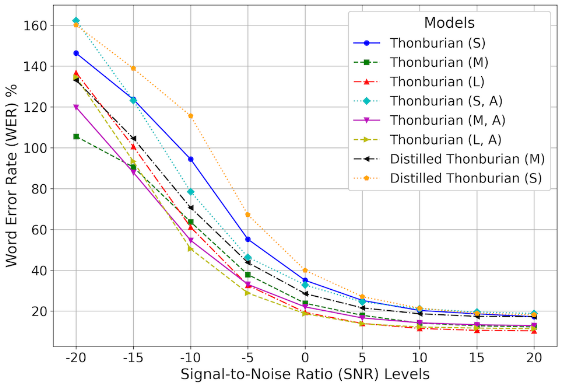


Figure 2: Robustness of the models under environmental noise.

scheme can enhance the adaptability of baseline Whisper models.

3.3 Model robustness under environmental noise

Augmented Thonburian Whisper large consistently outperforms all other models across different SNR levels of environmental noise corruption (Figure 2). From 0 to 20 dB, the non-augmented Thounburian models (S, M, L) show average WERs of 23.3, 16.18 and 13.15 respectively. The augmented models perform similarly under milder noise conditions with average WERs of 23.38, 15.83 and 13.56. As the noise corruptions become more severe (0 to -20 dB), the augmented variants outperform their counterparts. The standard deviations of WER for Thonburian Whisper small, medium and large are 5.10, 3.37, and 2.71 respectively. This suggests their performance is consistent across the SNR levels from 20 to 0, which is a good indicator of the model robustness. The proposed training scheme seems to have significantly improved the performance of the Whisper models under noisy conditions.

4 Conclusion

This study shows that Whisper based automatic speech recognition models can be successfully adapted and enhanced for Thai language. The proposed fine-tuning scheme and a combined corpus of Thai audios have led to substantial improvements in word error rate (WER) over existing baselines and previous works. Furthermore, we show that it is feasible to compress Whisper models through knowledge distillation with a fraction of data used in (Gandhi et al., 2023). This highlights the potential for creating lightweight yet strong ASR solutions for low resource languages. The strong performance of Thonburian Whisper models on the OOD financial dataset showcases their effectiveness and adaptability. This is notable considering the complex terminology, code switching tendencies and accented speech.

Limitations

The suggested audio augmentation methods can help create robust ASR systems designed for noisy environments. However, the differing levels of noise resistance among the models call for a further exploration of optimization tactics that can consistently improve robustness regardless of model sizes. The distilled models, especially the small one, are more adversely affected by noise corruptions. This indicates that there is still room for improvements in the distillation process to enhance robustness. Finally, while this study demonstrated the adaptability to financial domain data, further efforts are necessary to assess how well the models would work in a range of fields and situations for a language as complex and tonally varied as Thai.

Ethics Statement

This research complies with the ACL Ethics Policy[4]. All experiments were conducted using publicly available datasets, namely CMV13-Test, Google Fleurs Test, and Thai Central Dev, which are well-documented and widely accepted in the research community. We ensured that no personal or sensitive information was involved. We recognize the broader impacts of our work in automated speech recognition (ASR). While our models show significant improvements in ASR accuracy, it is essential to apply these advancements responsibly. We advocate for the ethical use of ASR technology to benefit diverse communities and prevent perpetuating biases or inequalities.

Acknowledgements

The authors acknowledge the NSTDA Supercomputer Center(ThaiSC) project #lT200142 for providing computing resources for this work.

---

[4] `https://www.aclweb.org/portal/content/acl-code-ethics`

Table 4: Computational Resources Comparison Across Distilled Models

Model	Encoder	Decoder	GPU Memory Usage	Memory Efficiency[1]	Short-Form Speed Up[2]	Long-Form Speed Up[3]
Thonburian (S)	12	12	461MB	-	-	-
Thonburian (M)	24	24	1,420MB	-	-	-
Distilled Thonburian (S)	12	4	317MB	1.45x	1.26x	1.72x
Distilled Thonburian (M)	24	4	816MB	1.74x	3.89x	2.41x

[1] Memory efficiency indicates the relative GPU memory usage effectiveness in FP16.
[2] Short-Form speed up is the time taken to transcribe approximately 6 seconds of audio.
[3] Long-Form speed up refers to the time taken to transcribe approximately 60 seconds of audio.

References

Ahmad A. M. Abushariah, Hua-Nong Ting, Mumtaz Begum Peer Mustafa, Anis Salwa Mohd Khairuddin, Mohammad A. M. Abushariah, and Tien-Ping Tan. 2023. Bilingual Automatic Speech Recognition: A Review, Taxonomy and Open Challenges. *IEEE Access*, 11:5944–5954. Conference Name: IEEE Access.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 58–68, Minneapolis, Minnesota. Association for Computational Linguistics.

Ekapol Chuangsuwanich, Atiwong Suchato, Korrawe Karunratanakul, Burin Naowarat, Chompakorn CChaichot, Penpicha Sangsa-nga, Thunyathon Anutarases, Nitchakran Chaipojjana, and Yuatyong Chaichana. 2020. Gowajee Corpus. Technical report, Chulalongkorn University, Faculty of Engineering, Computer Engineering Department.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Changhan Wang, Jeff Wang, and Skyler Wang. 2023. SeamlessM4T-Massively Multilingual & Multimodal Machine Translation. ArXiv:2308.11596 [cs].

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Sanchit Gandhi, Patrick von Platen, and Alexander M Rush. 2023. Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling. *arXiv preprint arXiv:2311.00430*.

Manraj Singh Grover, Pakhi Bamdev, Ratin Kumar Brala, Yaman Kumar, Mika Hama, and Rajiv Ratn Shah. 2021. audino: A Modern Annotation Tool for Audio and Speech. ArXiv:2006.05236 [cs, eess].

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech 2020*, pages 5036–5040. ISCA.

Rishabh Jain, Andrei Barcovschi, Mariam Yiwere, Peter Corcoran, and Horia Cucu. 2023. Adaptation of Whisper models to child speech recognition. ArXiv:2307.13008 [cs, eess].

Iver Jordal, Araik Tamazian, Emmanouil Theofanis Chourdakis, Céline Angonin, Tushar Dhyani, askskro, Nikolay Karpov, Omer Sarioglu, Baker-Bunker, kvilouras, Enis Berk Çoban, Florian Mirus, Jeong-Yoon Lee, Kwanghee Choi, MarvinLvn, SolomidHero, and Tanel Alumäe. 2023. iver56/audiomentations: v0.33.0.

Rakpong Kittinaradorn, Titipat Achakulvisut, Korakot Chaovavanich, Kittinan Srithaworn, Pattarawat Chormai, Chanwit Kaewkasi, Tulakan Ruangrong, and Krichkorn Oparad. 2019. DeepCut: A Thai word tokenization library using Deep Neural Network. Version Number: 1.0.

Burin Naowarat, Thananchai Kongthaworn, Korrawe Karunratanakul, Sheng Hui Wu, and Ekapol Chuangsuwanich. 2021. Reducing spelling inconsistencies in code-switching asr using contextualized ctc loss. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6239–6243.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, pages 2613–2617. ArXiv:1904.08779 [cs, eess, stat].

Wannaphong Phatthiyaphaibun, Chompakorn Chaksangchaichot, Peerat Limkonchotiwat, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. Thai Wav2Vec2.0 with CommonVoice V8. ArXiv:2208.04799 [cs, eess].

Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 1015–1018, New York, NY, USA. Association for Computing Machinery. Event-place: Brisbane, Australia.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518. PMLR. ISSN: 2640-3498.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20, pages 1–16, Atlanta, Georgia. IEEE Press.

Artit Suwanbandit, Jaturong Chitiyaphol, Sutthinan Chuenchom, Kanyarat Kwiecien, Husen Sawal, Ruslan Uthai, Orathai Sangpetch, and Ekapol Chuangsuwanich. 2023. Thai-dialect: Low resource thai dialectal speech to text corpora. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Silero Team. 2021. Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. Publication Title: GitHub repository.

Dual-Task Learning for AI-Generated Medical Text Detection and Named Entity Recognition

Saja Al-Dabet[1], Ban Alomar[1], Sherzod Turaev[1], Abdelkader Nasreddine Belkacem[2]
[1] Department of Computer Science and Software Engineering,
[2] Department of Computer and Network Engineering,
United Arab Emirates University
{700039885, 700039223, sherzod, belkacem}@uaeu.ac.ae

Abstract

The integration of artificial intelligence (AI) into the medical field has revolutionized documentation and diagnosis. However, the detection of AI-generated text within medical records remains a crucial task. This paper describes a dual-task learning framework using the ELECTRA model for detecting AI-generated medical texts and performing named entity recognition (NER). The dual-task model includes a binary classification head for identifying AI-generated texts and an NER head for extracting medical entities. Experiments on radiology report and medical texts datasets show that the proposed approach achieves robust performance, with F1 scores of 0.985 and 0.996 for classification and 0.51 and 0.68 for NER. The model achieves a high accuracy of 0.996 for medical text classification and 0.985 for MiMic classification, enhancing automated medical text analysis and supporting clinical decision-making.

1 Introduction

The advent of large language models such as ChatGPT (Generative Pretrained Transformer) has revolutionized various sectors (Radford et al., 2018), including the medical field, by enabling the generation of coherent and human-like text (Hamad et al., 2024; Hireche and Belkacem, 2024; Hireche et al., 2023; Jamil et al., 2024). These advances have facilitated tasks such as automated report generation, clinical documentation, and medical information dissemination. However, the spread of artificial intelligence (AI)-generated text in medicine raises significant concerns regarding the accuracy, reliability, and authenticity of the information contained therein. Misleading AI-generated medical content can have severe consequences, potentially compromising patient care and medical research integrity. Human medical writers, with their depth of understanding and expertise in the medical field, cannot at present be fully replaced by ChatGPT (Homolak, 2023; Liao et al., 2023; Tan et al., 2024). Additionally, there are concerns regarding potential bias in AI-generated content and the necessity for transparency in AI usage. This makes it essential to ensure the integrity and accuracy of medical information, indicating the important role of human oversight in creating medical content (Sajid and ul Hassan, 2022).

Distinguishing between human-written and AI-generated medical texts is challenging and requires robust detection methods. There are several differences between medical texts authored by humans and those generated by AI agents. Human-written texts have a larger vocabulary, greater diversity, and include specific information and numbers, making them detailed and contextually rich. In contrast, AI-generated texts use more common terminology, emphasizing fluency and logical structure, and are generally more neutral and positive in sentiment. In terms of parts-of-speech, AI-generated texts contain more nouns, determiners, plural nouns, and coordinating conjunctions, indicating a structured style, whereas humans use more cardinal digits and adverbs, reflecting greater specificity. Similarly, dependency parsing in AI-generated texts includes more determiners and conjuncts, with human texts having more numeric and adverbial modifiers. Furthermore, text perplexity is lower for AI-generated texts due to the replication of common patterns, whereas human texts display a greater degree of variation (Liao et al., 2023). Existing approaches, such as linguistic feature analysis and machine learning models, have shown promise, but often fall short in handling the complexities of medical language. To address these limitations, we propose a multitask model that leverages the capabilities of the ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al., 2020) language model. ELECTRA achieves superior performance in various natural language processing (NLP) tasks due to its efficiency in text encoding and understanding. Our proposed model utilizes ELECTRA for two primary tasks: differentiating between human-written and AI-generated texts and enhancing text comprehension through named entity recognition (NER). By integrating these tasks, the proposed model not only identifies AI-generated content, but also improves the understanding of medical texts, thereby increasing the accuracy of detection.

The integration of NER into the detection framework enables the model to identify and classify essential medical entities, thereby offering deeper insights into the context and content of the text. This dual-task approach ensures comprehensive analysis, capturing subtle differences between human- and AI-generated medical texts that may be overlooked by single-task models. Moreover, the enhanced text understanding provided by NER aids in the detection of inconsistencies and anomalies indicative of AI-generated content. This approach enhances parameter efficiency by sharing model parameters across tasks and leverages transfer learning, thereby allowing knowledge from one task to benefit the other. In addition, our model produces consistent predictions while simplifying deployment by reducing the need for separate models.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the previous literature, before Section 3 describes the proposed methodology used to develop the model. Section 4 covers the experiments and results. Finally, Section 5 concludes the paper.

2 Literature Review

In this section, recent papers on both AI-text detection and medical NER tasks are summarized.

2.1 AI-text detection task

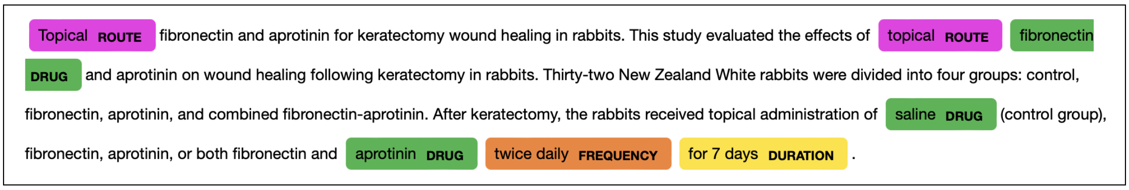
Guo et al. (2023) evaluated ChatGPT's performance in mimicking human expert responses using the Human ChatGPT Comparison Corpus (HC3), which includes around 40,000 questions and answers from both human experts and ChatGPT across various domains. The study utilized RoBERTa and GLTR models to analyze the text, revealing that RoBERTa significantly outperformed GLTR. Specifically, RoBERTa achieved F1 scores of 99.82% in full-text detection and 87.17% in sentence-level detection, compared with GLTR's 98.31% and 70.91%, respectively.

Scheibe and Mandl (2023) explored the effectiveness of models in distinguishing between human-written and machine-generated texts. Their study was framed within the AuTexTification 2023 shared task, focusing on automated text identification. The methodology uses the pre-trained DeBERTaV2 model (He et al., 2020), selected for its capabilities in handling text classification tasks, and a dataset that comprises a balanced mix of human and machine-generated texts, resulting in a robust training environment. In terms of results, the DeBERTaV2 model achieved a macro-F1 score of 67.2%, ranking 15th out of 76 submissions for subtask 1.

Verma et al. (2023) introduced Ghostbuster, developed by UC Berkeley researchers to detect AI-generated text. Ghostbuster uses the GPT-3 Davinci configuration to extract probabilistic features, and employs a linear classifier to identify machine-generated text. Token probabilities from the text-generating AI are not required, making Ghostbuster effective even with complex models. Tested on three datasets covering student essays, creative writing, and news articles, Ghostbuster achieved a 99% F1 score, outperforming models including DetectGPT and GPTZero.

Alamleh et al. (2023) explored machine learning-based approaches to detect ChatGPT-generated text. The authors evaluated their models on a Kaggle dataset of 10,000 instances, half from human sources and half generated by GPT-3.5. They employed a variety of machine learning and deep learning algorithms, including random forests, logistic regression, decision trees, support vector machines, AdaBoost, bagging classifiers, multilayer perceptrons, and long short-term memory (LSTM) networks, with a special focus on the extremely randomized trees classifier for its robustness in handling random data points. Their methodology involves sentence vectorization using the term frequency–inverse document frequency (TF-IDF) followed by classification. The highest achieved accuracy for distinguishing between human- and ChatGPT-generated texts was 77%.

Mitrović et al. (2023) investigated the ability of machine learning to detect AI-generated short online reviews, comparing a Transformer-based model with a perplexity-based approach. Two datasets were created: one with ChatGPT-generated texts from custom prompts and another with rephrased human-written reviews. The Shap-



Topical ROUTE fibronectin and aprotinin for keratectomy wound healing in rabbits. This study evaluated the effects of topical ROUTE fibronectin DRUG and aprotinin on wound healing following keratectomy in rabbits. Thirty-two New Zealand White rabbits were divided into four groups: control, fibronectin, aprotinin, and combined fibronectin-aprotinin. After keratectomy, the rabbits received topical administration of saline DRUG (control group), fibronectin, aprotinin, or both fibronectin and aprotinin DRUG twice daily FREQUENCY for 7 days DURATION .

Figure 1: Medical NER tagging example using en-core-med7-lg pretrained model.

ley additive explanations were used to identify influential features. The Transformer-based model proved more effective, achieving up to 98% accuracy for straightforward AI-generated texts and 79% for rephrased texts.

Liao et al. (2023) highlighted the risks of AI-generated content in medical contexts. The authors constructed datasets of both human-written and ChatGPT-generated medical texts, before analyzing the linguistic properties and employing machine learning to identify AI-generated content. The key differences were found to be the more detailed and varied human texts versus more general and fluent AI texts. A BERT-based model achieved an F1 score of over 95% in identifying AI-generated texts.

2.2 Medical NER tasks

For medical NER tasks, several studies have targeted different languages using machine and deep learning approaches. Review articles have specifically addressed medical and clinical NER approaches (Ahmad et al., 2023; Pagad and Pradeep, 2022).

Gaschi et al. (2023) evaluated cross-lingual transfer (CLT) and translation-based methods for medical NER in English, French, and German. They used the N2C2, MedNERF, and GERNERMED datasets, and applied fine-tuned multilingual models (XLM-R, mBERT) to N2C2 for CLT, as well as translating N2C2 for training (translate-train) and testing (translate-test). CLT with the XLM-R base yielded F1 scores of 79.1% for French and 72.2% for German. The translate-train method achieved F1 scores of 78.6% for French and 74.8% for German, while DrBERT PubMed scored 78.8% for French and 75% for German.

Park et al. (2023) developed a web service using BioBERT to integrate NER and relation extraction (RE) in the biomedical domain. The BioBERT base was fine-tuned using the NCBI Disease Corpus and BC2GM Corpus (for NER) and the Genetic Association Database (for RE). The NER system demonstrated high performance, achieving a precision of 85.16%, recall of 83.65%, and an F1 score of 84.4% for gene/protein recognition, and 89.04%, 89.69%, and 89.36%, respectively, for disease recognition. The Django-based web service allows users to input PubMed IDs, retrieve abstracts, and view color-coded NER results and interactive RE graphs.

Xu et al. (2018) presented a combined deep learning approach for medical NER. Utilizing datasets from the 2010 i2b2/VA NLP Challenges, their study implemented an attention-based LSTM architecture combined with a conditional random field to target document-level global information. This method employs pretrained word embeddings and bidirectional language models trained on the MIMIC-III corpus, and addresses the limitations of sentence-level NER by incorporating global context through neural attention mechanisms. The model achieved an impressive micro-F1 score of 85.71%.

Naseem et al. (2021) constructed BioALBERT, a domain-specific language model optimized for biomedical NER. The model was trained on large-scale biomedical corpora from PubMed and PMC, addressing the limitations of existing models through techniques such as factorized embedding parameterization, cross-layer parameter sharing, and sentence-order prediction. BioALBERT demonstrated significant performance improvements across various datasets: 7.47% for NCBI Disease, 10.63% for BC5CDR-Disease, 4.61% for BC5CDR-Chem, 3.89% for BC4CHEMD, 12.25% for BC2GM, 6.42% for JNLPBA, 6.19% for LINNAEUS, and 23.71% for Species-800.

Košprdić et al. (2023) proposed a biomedical NER approach using zero- and few-shot learning with six public corpora: CDR, CHEMDNER, BioRED, NCBI Disease, JNLPBA, and N2C2. They fine-tuned the BioBERT and PubMedBERT models, converting multiclass token classification into binary token classification to recognize unseen entity classes through semantic similarities from pretraining. The method achieved average F1 scores of 35.44% for zero-shot NER, 50.10% for one-shot NER, 69.94% for 10-shot NER, and 79.51% for 100-shot NER.

3 Proposed Model

This section describes the proposed model. Multitask learning is used to train a single model on multiple tasks simultaneously, improving both generalization and performance through shared representations. This approach enhances parameter efficiency by sharing model parameters across tasks, which is beneficial when there are limited computational resources or datasets. Transfer learning leverages knowledge from one task to enhance another. Multitask models yield consistent and coherent predictions, simplifying deployment by reducing the need for separate models. In a dual-task model for text classification and NER tagging, shared linguistic and entity recognition capabilities enhance the overall performance. The following subsections detail the proposed architecture. Figure 2 provides an overview.

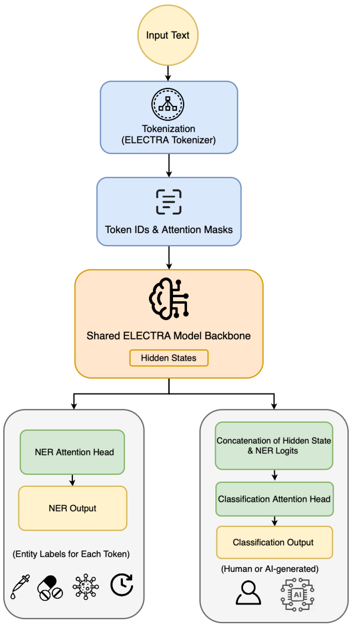

Input Text
Tokenization
(ELECTRA Tokenizer)
Token IDs & Attention Masks
Shared ELECTRA Model Backbone
Hidden States
NER Attention Head
NER Output
(Entity Labels for Each Token)
Concatenation of Hidden State & NER Logits
Classification Attention Head
Classification Output
(Human or AI-generated)
AI

Figure 2: Proposed model architecture.

3.1 Data preprocessing

Data preprocessing makes a significant contribution to model performance. The following preprocessing steps are implemented:

- *Identification of null entries*: an initial assessment is performed to identify columns containing missing values.
- *Tokenization*: text data are tokenized to convert sentences into single tokens. This process is essential for subsequent text processing and model input preparation.
- *Remove special characters*: special characters that do not add to the semantic meaning of the text are removed. This step helps in cleaning the data and minimizing noise.
- *NER data annotation*: the data are annotated with medical NER tags using the SpaCy pretrained model ("en-core-med7-lg" version) (AI, 2024). This model is specifically designed for medical NER, identifying entities such as DRUG (names of medications), DOSAGE (dosage information and units), DURATION (duration of medication use or treatment), FORM (medication forms, i.e., tablets or injections), FREQUENCY (how often a medication is taken), ROUTE (route of administration, i.e., oral or intravenous), and STRENGTH (strength of the medication). The use of these NER tags ensures the precise identification and categorization of relevant medical entities within the text. Figure 1 illustrates an example of an annotated medical report tagged using the "en-core-med7-lg" pretrained model.
- *Encoding extraction*: encoding vectors and labels are extracted from the tokenized datasets for both classification and NER tasks. This involves generating numerical representations of the text data that are suitable for model training.
- *Label padding and conversion for NER*: NER labels are padded and converted from string tuples to integer labels using a label map. This ensures that the labels have a consistent format and are aligned with the input sequences, which is necessary for effective model training.

3.2 Framework architecture

To overcome the challenges of distinguishing between human-written and AI-generated medical texts, a multitask framework leveraging the ELECTRA language model is proposed. This framework is designed to perform two primary tasks simultaneously: differentiating between human- and AI-generated texts and enhancing text comprehension through NER. By integrating these tasks, the model not only enhances the accuracy with which AI-generated content is detected, but also provides a deeper understanding of the context and content of the medical texts. ELECTRA (Clark et al., 2020) represents a pretraining approach for text encoders, diverging significantly from traditional masked language modeling methods such as BERT (Devlin et al., 2018). Rather than masking random tokens and predicting their original forms, ELECTRA modifies the input by replacing specific tokens with plausible alternatives produced by a smaller auxiliary network, known as the generator.

The primary model (discriminator) is then tasked with identifying whether each token in the modified input is original or has been replaced. This replaced token detection strategy leverages the entire input sequence, thereby enhancing both sample efficiency and computational effectiveness. The discriminator plays a critical role, as it learns to differentiate between authentic tokens and those introduced by the generator using the full context of the input data. This discriminative task not only improves the training efficiency, but also enhances performance on downstream tasks. The architecture of ELECTRA integrates both a generator and a discriminator, resulting in superior results with fewer computational resources (Hao et al., 2021; Ozyurt, 2020). Algorithm 1 in the appendix describes the proposed dual-task learning process for both classification and NER tasks.

4 Experiments and Results

4.1 Dataset

The medical dataset utilized in this study comprises two primary components, as described by Liao et al. (2023). The medical abstract dataset is sourced from a publicly available Kaggle dataset (Kamath, 2023) and includes texts related to five medical conditions: digestive system diseases, cardiovascular diseases, neoplasms, nervous system diseases, and general pathological conditions. The radiology report dataset, which is based on the work of Johnson et al. (2016), includes selected radiology reports. A total of 4400 samples were obtained from both the radiology report and medical abstract datasets as human-written medical texts. To create corresponding ChatGPT-generated texts, a text continuation method was applied, resulting in datasets containing 8800 samples each for the medical abstracts and radiology reports. Both datasets were then divided into 70% for training, 10% for validation, and 20% for testing subsets, yielding 3080 samples for training, 440 for validation, and 880 for testing in each dataset.

4.2 Evaluation metrics

To evaluate the performance of the proposed model, a comprehensive set of evaluation metrics was employed. The precision, recall, and F1 score are essential metrics in the context of distinguishing between AI-generated and human-written medical texts.

4.3 Experimental settings

The experiments were conducted on the Kaggle platform using the GPU-enabled feature. The applied model based on the "electra-small-discriminator" checkpoint and tokenization was handled by the ElectraTokenizer layer. For classification tasks involving both the MiMic and medical datasets, the batch size for classification tasks was set to 16, whereas for NER tasks, it was set to 8. NER tasks utilize seven labels, while classification tasks are binary, involving two labels. The AdamW optimizer was used (Loshchilov and Hutter, 2017) with a learning rate of 5×10^{-5}. The training process involved separate head optimization with five epochs for both the classification and NER heads, followed by joint optimization epochs.

4.4 Results and discussion

4.4.1 Evaluating dual-task performance

Table 1 compares the proposed model with other models from the literature. The proposed model performs robustly across all metrics for both NER and classification tasks, outperforming ELECTRA, RoBERTa (Liu et al., 2019), BioBERT (Lee et al., 2020), and XLNet (Yang et al., 2019), and surpassing the baseline model of Liao et al. (2023). Four main models were used: Perplexity-CLS, CART, XGBoost, and BERT. For Perplexity-CLS, BioGPT calculates the text perplexity, with the optimal threshold identified using the validation set. The CART model uses TF-IDF for vectorization, a decision tree with a maximum depth of four, and the Gini impurity for feature division. The XGBoost model also uses TF-IDF and sets the maximum depth for base learners to four. The BERT model achieves the best performance due to its advanced text processing capabilities.

Table 1: Performance evaluation for models.

Classification Task								
Model	Medical text				MiMic			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Perplexity-CLS (Liao et al., 2023)	-	0.728	0.724	0.723	-	0.831	0.828	0.828
CART (Liao et al., 2023)	-	0.777	0.745	0.738	-	0.829	0.825	0.824
XGBoost (Liao et al., 2023)	-	0.898	0.893	0.893	-	0.899	0.898	0.898
BERT (Liao et al., 2023)	-	0.958	0.958	0.958	-	0.968	0.967	0.967
BioBERT	0.948	0.943	0.942	0.944	0.970	0.968	0.968	0.969
RoBERTa	0.999	0.999	0.999	0.999	0.989	0.990	0.989	0.989
XLNet	0.998	0.998	0.998	0.998	0.988	0.988	0.988	0.988
ELECTRA	0.996	0.996	0.996	0.996	0.985	0.985	0.986	0.985
Named Entity Recognition Task								
RoBERTa	0.54	0.41	0.47	0.58	0.75	0.72	0.72	0.73
XLNet	0.72	0.42	0.48	0.51	0.45	0.31	0.36	0.38
ELECTRA	0.68	0.45	0.51	0.56	0.93	0.91	0.91	0.92

In comparison with the other models considered in this study, ELECTRA demonstrates faster and more robust performance. The ELECTRA model utilizes a pretraining method that is more computationally efficient than the traditional masked language modeling employed by models such as BERT. Instead of masking and predicting random tokens, ELECTRA modifies the input by replacing some tokens with plausible alternatives generated by a small auxiliary network, and then trains a discriminator to determine whether each token is original text or substituted text. For the classification task, ELECTRA demonstrates robust performance on both the medical text and MiMic datasets. Specifically, ELECTRA achieves an accuracy of 0.985 for the MiMic dataset, with precision and recall scores of 0.985 and 0.986, respectively, resulting in an F1 score of 0.985. For the medical texts, ELECTRA achieves an accuracy of 0.996, with precision, recall, and F1 scores all at 0.996. This performance is comparable to, and in some cases exceeds, that of other transformer-based models such as BioBERT. The high F1 scores indicate that ELECTRA is highly effective at differentiating between AI-generated and human-written texts, making it a strong candidate for this classification task.

In the NER task, ELECTRA produces balanced performance across the datasets. On the MiMic dataset, ELECTRA achieves an accuracy of 0.93, precision and recall of 0.91, and an F1 score of 0.92. The medical reports dataset, however, presents a more challenging environment for the model due to the nature of the written text. ELECTRA achieves an accuracy of 0.68 and an F1 score of 0.56 on this dataset, with precision at 0.45 and recall at 0.51.

In the field of medical AI, the development of a stable architecture capable of both classification and NER tasks is essential. ELECTRA demonstrates efficient classification, achieving high F1 scores on both the medical text and MiMic datasets, thereby ensuring precise differentiation between AI-generated and human-written texts. Although there is potential for improvement in terms of NER performance, the ability of ELECTRA to identify and classify medical entities remains significant. This stability across multiple tasks enhances the reliability of automated medical text analysis, facilitating more accurate clinical decision-making and efficient information processing.

To evaluate the effect of using a dual-task model instead of a single classification model, the ELECTRA classification model was tested alone and achieved an accuracy of 0.967, precision of 0.968, recall of 0.967, and an F1 score of 0.967. Using the dual-task ELECTRA model, which integrates NER parameters, enhances the results over those given by the ELECTRA model alone. The integration of NER allows the model to better understand and classify complex medical texts by recognizing and categorizing relevant entities within the text, thus improving the overall accuracy and reliability of the classification.

The receiver operating characteristic (ROC) curves are shown in Fig. 3. These curves evaluate the performance of the classification and NER tasks on the medical and MiMic datasets. The top-left plot shows the overall ROC curves, with an area under the curve (AUC) of 1.0 for both datasets, indicating significant classification performance in distinguishing AI-generated from human-written text. The top-right plot displays the ROC curves for the NER task, with slightly better performance on the MiMic dataset than the medical dataset.

The bottom-left plot presents multiple ROC curves for the medical dataset's NER performance across different classes (0–6, representing form, dosage, route, frequency, drug, strength, and duration), with Class 4 (Drug) and Class 0 (Form) having the highest and lowest AUCs, respectively. Similarly, the bottom-right plot shows the MiMic dataset's NER performance, with Class 1 (Dosage) and Class 6 (Duration) being the most challenging and easiest classes, respectively. The micro-average curves in the bottom plots indicate good overall NER performance.

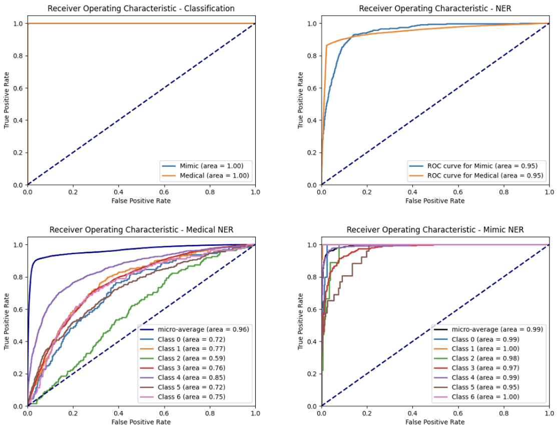


Figure 3: ROC curves for both experimented tasks.

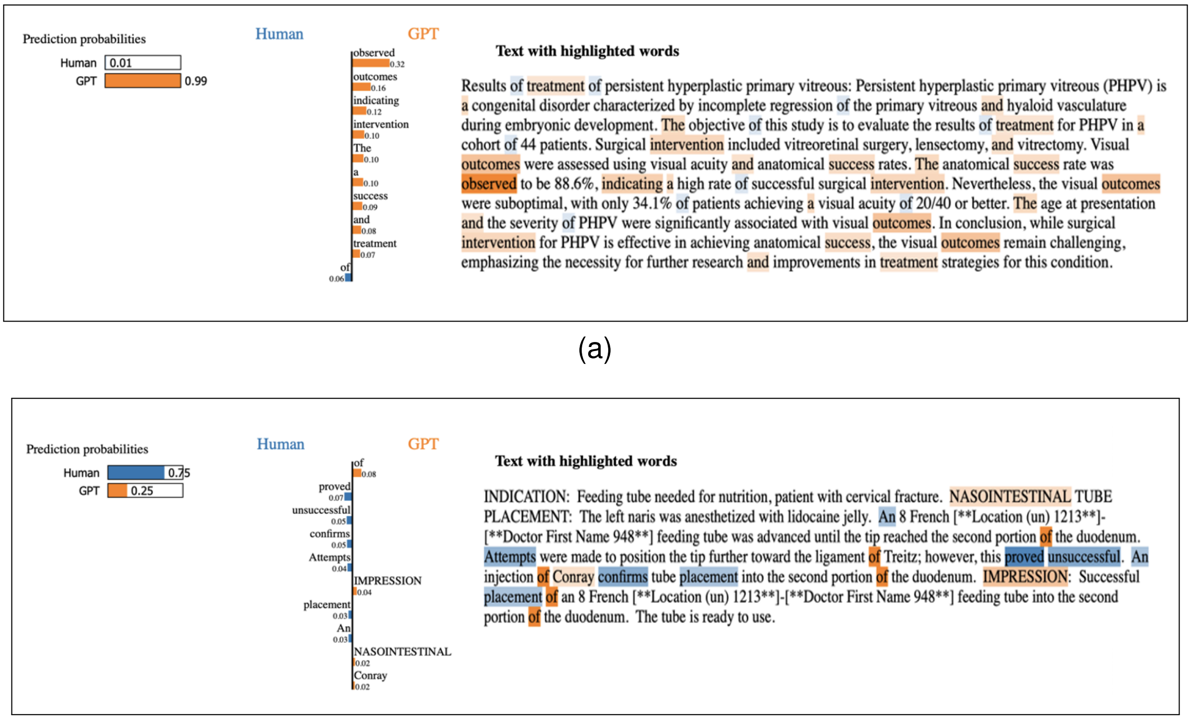
To add more explainability to the trained model, the Local Interpretable Model-agnostic Explanations (LIME) tool was utilized (Ribeiro et al., 2016). LIME increases interpretability by approximating the behavior of complex models around specific predictions. The trained weights of the ELECTRA model were used to plot these figures. In Fig. 4(a), the model assigns a high probability of 0.99 to the text being GPT-generated and only 0.01 to it being human-written, with key terms such as "treatment", "outcomes", "indicating", and "intervention" highlighted in orange, indicating their significant contribution to the model's classification decision. Figure 4(b) shows the prediction probability of 0.75 for the text being human-written, while the probability for GPT generation is 0.25, where key terms such as "proved", "unsuccessful", "confirms", "attempts", and "placement" are highlighted in blue, indicating their significant contribution to the human-written classification. In contrast, the terms "of" and "Conray" are highlighted in orange, showing their association with the GPT-generated classification. The resulting predictions are both correct.

To evaluate the effect of using Transformer models on the NER task alone, additional explorations were conducted. ClinicalBERT (Huang et al., 2019), SciBERT (Beltagy et al., 2019), BlueBERT (Peng et al., 2019), and BioBERT are variants of the BERT model (Devlin et al., 2018) tailored for medical and clinical usage. ClinicalBERT is pretrained on clinical notes and medical records, enhancing its effectiveness in healthcare-related tasks. SciBERT is pretrained on scientific literature from Semantic Scholar, making it suitable for scientific and academic applications. BlueBERT is trained on a combination of biomedical and clinical texts, specifically PubMed abstracts and MIMIC-III clinical notes, allowing it to handle both domains proficiently. BioBERT is pretrained on extensive biomedical literature, including PubMed abstracts and full-text articles from PubMed Central, resulting in optimization for understanding biomedical texts. Other models such as BERT, RoBERT, and ALBERT (Lan et al., 2019) were also included in this experiment. Table 2 presents the results obtained using these Transformers for the NER task.

Table 2: NER task evaluation.

Model	**Dataset**	**Acc**	**Precision**	**Recall**	**F1-score**
ClinicalBERT	Medical	0.93	0.80	0.76	0.78
	MiMic	0.99	0.98	0.99	0.99
SciBERT	Medical	0.88	0.70	0.63	0.65
	MiMic	0.95	0.95	0.93	0.94
BlueBERT	Medical	0.91	0.77	0.69	0.72
	MiMic	0.97	0.98	0.96	0.97
BioBERT	Medical	0.91	0.78	0.71	0.74
	MiMic	0.97	0.97	0.96	0.96
ELECTRA	Medical	0.88	0.70	0.57	0.63
	MiMic	0.97	0.97	0.96	0.96
BERT	Medical	0.90	0.75	0.69	0.72
	MiMic	0.99	0.99	0.99	0.99
RoBERT	Medical	0.86	0.66	0.48	0.55
	MiMic	0.93	0.92	0.87	0.89
ALBERT	Medical	0.86	0.54	0.41	0.47
	MiMic	0.62	0.26	0.26	0.26

Comparative analysis of the pretrained Transformer-based models for NER tasks across the medical and MiMic datasets reveals significant performance variability. ClinicalBERT and BERT demonstrate exceptional proficiency, achieving the highest F1 scores of 0.99 on the MiMic dataset and 0.78 on the medical dataset. This outstanding performance can be attributed to their architecture, which enhances their ability to accurately identify and classify named entities. SciBERT and ELECTRA achieve moderately good performance on the medical dataset (F1 scores of 0.65 and 0.63), but perform strongly on the MiMic



Results of treatment of persistent hyperplastic primary vitreous: Persistent hyperplastic primary vitreous (PHPV) is a congenital disorder characterized by incomplete regression of the primary vitreous and hyaloid vasculature during embryonic development. The objective of this study is to evaluate the results of treatment for PHPV in a cohort of 44 patients. Surgical intervention included vitreoretinal surgery, lensectomy, and vitrectomy. Visual outcomes were assessed using visual acuity and anatomical success rates. The anatomical success rate was observed to be 88.6%, indicating a high rate of successful surgical intervention. Nevertheless, the visual outcomes were suboptimal, with only 34.1% of patients achieving a visual acuity of 20/40 or better. The age at presentation and the severity of PHPV were significantly associated with visual outcomes. In conclusion, while surgical intervention for PHPV is effective in achieving anatomical success, the visual outcomes remain challenging, emphasizing the necessity for further research and improvements in treatment strategies for this condition.

(a)

INDICATION: Feeding tube needed for nutrition, patient with cervical fracture. NASOINTESTINAL TUBE PLACEMENT: The left naris was anesthetized with lidocaine jelly. An 8 French [**Location (un) 1213**]-[**Doctor First Name 948**] feeding tube was advanced until the tip reached the second portion of the duodenum. Attempts were made to position the tip further toward the ligament of Treitz; however, this proved unsuccessful. An injection of Conray confirms tube placement into the second portion of the duodenum. IMPRESSION: Successful placement of an 8 French [**Location (un) 1213**]-[**Doctor First Name 948**] feeding tube into the second portion of the duodenum. The tube is ready to use.

(b)

Figure 4: Explainable examples using LIME. (a) Medical text example. (b) MiMic text example.

dataset (F1 scores of 0.94 and 0.96). BlueBERT and BioBERT consistently perform well across both datasets, achieving F1 scores of 0.72 and 0.74 on the medical dataset and 0.97 and 0.96 on the MiMic dataset. RoBERTa and ALBERT display weak performance on the medical dataset (F1 scores of 0.55 and 0.47), with ALBERT underperforming on the MiMic dataset (F1 score of 0.26).

In multitask learning, the separate tasks can influence each other's outcomes. In a dual-task setup with text classification and NER, classification is often improved by joint training. This is due to shared representations capturing general features that are useful for both tasks, with NER enhancing the model's linguistic and semantic understanding. This positive transfer acts as regularization, reducing overfitting and boosting classification performance. However, NER might perform better alone due to task interference and complexity in balancing losses in a dual-task model. Thus, while multitask learning benefits classification, it poses challenges for optimizing both tasks.

5 Conclusion and Future Work

This study developed a dual-task learning framework using the ELECTRA model to detect AI-generated medical texts and perform NER. The integrated approach, combining a binary classification head and an NER head, showed robust performance across medical text and radiology report datasets. The framework effectively distinguishes human-written from AI-generated texts and extracts critical medical entities, enhancing detection accuracy and text comprehension. Experiments demonstrated that the ELECTRA model outperforms others in terms of inference speed and prediction robustness, achieving high F1 scores for both classification and NER tasks.

Future work will attempt to extend and refine the proposed framework by exploring additional datasets and domains to evaluate the model's generalizability and robustness across various types of medical texts. Moreover, incorporating more advanced techniques for handling complex medical terminology and context-specific nuances could further improve the framework's performance and applicability in real-world scenarios.

Acknowledgments

This research was partially funded by the United Arab Emirates University (UAEU) under Grant UPAR-12T055, with additional support from the UAEU Strategic Research Grant G00003676 through the Big Data Analytics Center, and the UAEU SURE+ Grant G00004748.

References

Pir Noman Ahmad, Adnan Muhammad Shah, and KangYoon Lee. 2023. A review on electronic health record text-mining for biomedical name entity recognition in healthcare domain. In *Healthcare*, volume 11, page 1268. MDPI.

Explosion AI. 2024. spacy models: en_core_web_md. Version 3.5.0.

Hosam Alamleh, Ali Abdullah S AlQahtani, and AbdElRahman ElSaid. 2023. Distinguishing human-written and chatgpt-generated text using machine learning. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pages 154–158. IEEE.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Félix Gaschi, Xavier Fontaine, Parisa Rastin, and Yannick Toussaint. 2023. Multilingual clinical ner: Translation or cross-lingual transfer? In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 289–311.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arxiv. *Preprint posted online on*, 18.

Zineb Touati Hamad, Nuraini Jamil, and Abdelkader Nasreddine Belkacem. 2024. Chatgpt's impact on education and healthcare: Insights, challenges, and ethical consideration. *IEEE Access*.

Yaru Hao, Li Dong, Hangbo Bao, Ke Xu, and Furu Wei. 2021. Learning to sample replacements for electra pre-training. *arXiv preprint arXiv:2106.13715*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Abdelhadi Hireche and Abdelkader Nasreddine Belkacem. 2024. Integrating pepper robot and gpt for neuromyth educational conversation. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–6. IEEE.

Abdelhadi Hireche, Abdelkader Nasreddine Belkacem, Sadia Jamil, and Chao Chen. 2023. Newsgpt: Chatgpt integration for robot-reporter. *arXiv preprint arXiv:2311.06640*.

Jan Homolak. 2023. Opportunities and risks of chatgpt in medicine, science, and academic publishing: a modern promethean dilemma. *Croatian Medical Journal*, 64(1):1.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Nuraini Jamil, Fahed Namir Saghir, Hassan Alshanqiti, Ali Khalifa Ali Almansoori, Abdulrahman Saeed, Ali Ahmad, and Abdelkader Nasreddine Belkacem. 2024. On combining the potential of social robots and chatgpt for enhanced learning. In *2024 12th International Conference on Information and Education Technology (ICIET)*, pages 226–231. IEEE.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Chaitanya Kamath. 2023. Medical text dataset. Accessed: 2024-06-04.

Miloš Košprdić, Nikola Prodanović, Adela Ljajić, Bojana Bašaragin, and Nikola Milosevic. 2023. From zero to hero: Harnessing transformers for biomedical named entity recognition in zero-and few-shot contexts. *Available at SSRN 4463335*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

W Liao, Z Liu, H Dai, S Xu, Z Wu, Y Zhang, X Huang, D Zhu, H Cai, T Liu, et al. 2023. Differentiate chatgpt-generated and human-written medical texts. arxiv 2023. *arXiv preprint arXiv:2304.11567*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

Usman Naseem, Matloob Khushi, Vinay Reddy, Sakthivel Rajendran, Imran Razzak, and Jinman Kim. 2021. Bioalbert: A simple and effective pre-trained language model for biomedical named entity recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Ibrahim Burak Ozyurt. 2020. On the effectiveness of small, discriminatively pre-trained language representation models for biomedical text mining. *bioRxiv*, pages 2020–05.

Naveen S Pagad and N Pradeep. 2022. Clinical named entity recognition methods: an overview. In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 2*, pages 151–165. Springer.

Yeon-Ji Park, Min-a Lee, Geun-Je Yang, Soo Jun Park, and Chae-Bong Sohn. 2023. Web interface of ner and re with bert for biomedical text mining. *Applied Sciences*, 13(8):5163.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Ujala Sajid and Faheem ul Hassan. 2022. Chatgpt and its effect on shaping the future of medical writing. *Pakistan Journal of Ethics*, 2(2):38–43.

Tatjana Scheibe and Thomas Mandl. 2023. Univ. of hildesheim at autextification 2023: Detection of automatically generated texts.

Songtao Tan, Xin Xin, and Di Wu. 2024. Chatgpt in medicine: prospects and challenges: a review article. *International Journal of Surgery*, pages 10–1097.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Guohai Xu, Chengyu Wang, and Xiaofeng He. 2018. Improving clinical named entity recognition with global neural attention. In *Web and Big Data: Second International Joint Conference, APWeb-WAIM 2018, Macau, China, July 23-25, 2018, Proceedings, Part II 2*, pages 264–279. Springer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

A Appendix

Algorithm 1 Multitask learning for text classification and NER.

```
Input:
$\mathcal{D} = \{(\mathbf{x}_i, y_i, \mathbf{Y}_i^{\text{NER}})\}_{i=1}^{N}$: Dataset
$\mathcal{M}$: Pretrained ELECTRA model
$\mathcal{T}$: Tokenizer
$L_{\text{max}}$: Maximum sequence length
$\mathcal{L}$: Label set for NER
$E_c, E_n$: Epochs for classification and NER pretraining
$E_j$: Epochs for joint training
Output: Trained multitask model $\mathcal{F}$
                                                    // Preprocessing
for $i = 1$ to $N$ do
    $\mathbf{x}_i^t \leftarrow \mathcal{T}(\mathbf{x}_i, L_{\text{max}})$
    $\mathbf{Y}_i^{\text{NER}} \leftarrow$ ConvertAndPad($\mathbf{Y}_i^{\text{NER}}, \mathcal{L}, L_{\text{max}}$)
end for
                                                    // Model architecture
$\mathcal{F}_{\text{BERT}} \leftarrow \mathcal{M}$
$\mathcal{F}_{\text{NER}} \leftarrow$ LinearLayer($d_{\text{BERT}}, |\mathcal{L}|$)
$\mathcal{F}_{\text{CLS}} \leftarrow$ LinearLayer($d_{\text{BERT}} + |\mathcal{L}|, 2$)
                                                    // Loss functions
$\mathcal{L}_{\text{CLS}} \leftarrow$ CrossEntropyLoss()
$\mathcal{L}_{\text{NER}} \leftarrow$ CrossEntropyLoss(ignore_index $= -1$)
                                                    // Separate pretraining
for $e = 1$ to $\max(E_c, E_n)$ do
    if $e \leq E_c$ then
        Train $\mathcal{F}_{\text{BERT}}$ and $\mathcal{F}_{\text{CLS}}$ using $\mathcal{L}_{\text{CLS}}$
    end if
    if $e \leq E_n$ then
        Train $\mathcal{F}_{\text{BERT}}$ and $\mathcal{F}_{\text{NER}}$ using $\mathcal{L}_{\text{NER}}$
    end if
end for
                                                    // Joint training
for $e = 1$ to $E_j$ do
    for $(\mathbf{x}_i^t, y_i, \mathbf{Y}_i^{\text{NER}})$ in $\mathcal{D}$ do
        $\mathbf{H}_i \leftarrow \mathcal{F}_{\text{BERT}}(\mathbf{x}_i^t)$
        $\mathbf{Z}_i^{\text{NER}} \leftarrow \mathcal{F}_{\text{NER}}(\mathbf{H}_i)$
        $\mathbf{h}_i^{\text{CLS}} \leftarrow \mathbf{H}_i[0, :]$
        $\mathbf{z}_i^{\text{CLS}} \leftarrow \mathcal{F}_{\text{CLS}}([\mathbf{h}_i^{\text{CLS}}; \mathbf{Z}_i^{\text{NER}}[0, :]])$
        $L_{\text{CLS}} \leftarrow \mathcal{L}_{\text{CLS}}(\mathbf{z}_i^{\text{CLS}}, y_i)$
        $L_{\text{NER}} \leftarrow \mathcal{L}_{\text{NER}}(\mathbf{Z}_i^{\text{NER}}, \mathbf{Y}_i^{\text{NER}})$
        Update $\mathcal{F}$ by minimizing $L_{\text{CLS}} + L_{\text{NER}}$
    end for
end for=0
```

Comparative Analysis of Modality Fusion Approaches for Audio-Visual Person Identification and Verification

Aref Farhadipour* and **Masoumeh Chapariniya** and **Teodora Vukovic** and **Volker Dellwo**
Department of Computational Linguistics, University of Zurich, Switzerland
`masoumeh.chapariniya@uzh.ch`
`teodora.vukovic2@uzh.ch`
`volker.dellwo@uzh.ch`

Abstract

Multimodal learning involves integrating information from various modalities to enhance learning and comprehension. We compare three modality fusion strategies in person identification and verification by processing two modalities: voice and face. In this paper, a one-dimensional convolutional neural network is employed for x-vector extraction from voice, while the pre-trained VGGFace2 network and transfer learning are utilized for face modality. In addition, gammatonegram is used as speech representation in engagement with the Darknet19 pre-trained network. The proposed systems are evaluated using the K-fold cross-validation technique on the 118 speakers of the test set of the VoxCeleb2 dataset. The comparative evaluations are done for single-modality and three proposed multimodal strategies in equal situations. Results demonstrate that the feature fusion strategy of gammatonegram and facial features achieves the highest performance, with an accuracy of 98.37% in the person identification task. However, concatenating facial features with the x-vector reaches 0.62% for EER in verification tasks.

1 Introduction

Biometric modalities encompass distinct static physiological traits that remain consistent within the human body, like fingerprints, as well as dynamic behavioral traits that are unique characteristics displayed in response to interactions with the environment, such as gait. However, some modalities can be a combination of both static and dynamic, such as speech (Minaee et al., 2023). In the real world, the human brain simultaneously processes multiple modalities to recognize the identity of each person. Although the exact mechanism of multimodal processing in the human brain remains unclear, the human mind can effortlessly identify individuals based on their faces and voices with minimal errors (Perrodin et al., 2015). Voice and face have garnered significant attention in the development of automatic identity recognition systems (Minaee et al., 2023; Farhadipour and Taghipour, 2023; Farhadipour and Veisi, 2024a). While both modalities can change over time or be vulnerable to spoof attacks, combining and analyzing mixed forms of these two modalities can potentially increase the uniqueness of biometric features for each person in identity recognition. In human-machine interaction, identity recognition plays a crucial role. It has been applied in various tasks such as access control, automatic monitoring of older individuals, rehabilitation programs for people with physical and mental disabilities, etc. Automatic identity recognition can be categorized into two main tasks: person verification and person identification. Person verification involves authenticating a claimed identity. This typically involves a two-class classification, comparing the claimed identity with unique specific and universal background models. In other words, the system determines whether to accept or reject the claimed identity. On the other hand, person identification refers to the process of identifying an identity in a multi-class classification scenario, where the system needs to determine the person's identity from a large pool of individuals who have been previously trained. While there has been significant research on developing multimodal person verification systems (Shah et al., 2023), there is still limited work done in the context of multimodal person identification.

In recent years, deep learning has emerged

---

*Correspondence: `aref.farhadipour@uzh.ch` as an effective approach to pattern recognition, allowing for the modeling of complex functions. This approach enables multimodal signal processing and can be implemented through different tasks. In this work, we utilized deep learning approaches in identity recognition tasks with three different strategies. The first strategy is sensor fusion, where raw information, such as sound and image data, are directly integrated into a classifier. This approach combines the information from different sources at the input level without front-end processing. The second strategy is feature fusion, which involves separately extracting low-dimensional features from each modality and then combining them to feed into a classifier. This approach allows for the extraction of modality-specific features before integration. The third strategy is score fusion, where separate classifiers are designed for each modality, and the final scores from each classifier are merged and fed into a decision-making system.

In this work, it is assumed that the person's voice and face information are accessible simultaneously. We trained two separate systems to learn and recognize each modality individually for person identification and verification. Furthermore, three multimodal strategies have been proposed, including all three discussed modes: sensor-level fusion, feature-level fusion, and score-level fusion in identification tasks. Finally, the best fusion mode that is explored in the identification task is utilized in the verification scenario.

To achieve these objectives, the pre-trained VGGFace2 (Cao et al., 2018) is utilized for the visual modality. However, for the voice modality, we proposed two methods consisting of gammatonegram representation (Farhadipour and Veisi, 2024b) and x-vector (Snyder et al., 2018). This approach results in the creation of the proposed FaceNet and VoiceNet models for each modality from the VoxCeleb2 dataset.

In the feature and score fusion modes, we mixed facial features with gammatonegram and x-vector separately, and a softmax layer was employed for multimodal learning by combining the information from both modalities. However, in score fusion mode, because we chose a two-dimensional space for mixing the modalities, we just concatenated facial features with gammatonegram representation. The proposed systems are trained and evaluated using the test section of the VoxCeleb2 dataset, which consists of 118 speakers. A K-fold cross-validation approach is utilized to ensure robustness and reliability.

The rest of the paper is organized as follows. Section 2 presents an overview of related works. In Section 3, the proposed strategies for multimodal learning in person identification and verification are described in detail. Section 4 focuses on the evaluation setup. Experimental results are reported in section 5. Discussing the results and comparison with previous works are done in section 6. Finally, section 7 concludes the work and discusses some ideas for future trends.

2 Related Works

Multimodal learning systems in identity identification integrated different modalities, including the fusion of fingerprint and DNA (Ramana et al., 2022), face and gait (Prakash et al., 2023), and face, palmprint and iris (Aldjia and Leila, 2021). In integrating face and speech, the main works focused on speaker verification (Wang et al., 2022), and few works are accomplished in speaker identification. It should be noted that the existence of the annual VoxSRC challenge had a great impact on this tendency (Huh et al., 2023). In this part, we report some of the previous works in multimodal speaker identification.

In the paper (Chung et al., 2018), an EER of 4.42% was reported on the Voxceleb1 dataset in a speaker verification task. The same authors in another article (Nagrani et al., 2020) reported 2.95% as EER on the Voxceleb1 dataset. Authors in paper (Moufidi et al., 2023) utilized a residual neural network to encode depth videos, while a time delay neural network architecture was used to encode voice signals. To evaluate the performance, 1,000 random speakers from the VoxCeleb2 dataset were selected. In that work, the accuracy of the three systems based on voice, depth of the lip regions, and overall fusion reached 56.03%, 17.49%, and 64.11%, respectively.

Alam et al., (Alam et al., 2015) introduced a confidence-based score fusion framework for audio-visual biometric identification. They proposed the confidence-ratio approach. The researchers utilized a database consisting of 43 individuals. The results indicated that, in the best case with clean data, an accuracy of 90.45% was achieved. Paper (Asadpour et al., 2011) employed a combination of audio and video features by implementing a hidden Markov model.

The study (Shah et al., 2023) suggested a two-branch network to extract facial and voice signal features, using a support vector machine to classify speakers based on single and multi-domain features. Their method got an accuracy of 97.2 % for Speaker identification performance on VoxCeleb1. In (Sarı et al., 2021), a multi-view model with a shared classifier to map audio and video into a unified space was introduced. The unimodal and audio-visual fusion approaches achieved an equal error rate of 1.6% on the VoxCeleb2 dataset in the person verification.

Qian et al.,(Qian et al., 2021) introduced three types of audio-visual deep neural networks: feature level, embedding level, and embedding level combination with joint learning. The VoxCeleb2 test trial list experiments demonstrated equal error rates of 5.08% and 2.89% for visual and audio modality systems, respectively. Tao et al., (Tao et al., 2020) employed an audio-visual cross-modal discrimination network for speaker recognition. The best setting of the system gained an accuracy of 86.12% on VoxCeleb2 for speaker identification in score-level fusion.

The paper (Stefanidi et al., 2020) introduced an approach to person identification using CNNs on the VoxCeleb1 audiovisual database. The outcomes achieved an 86.97% top-5 accuracy. Cai et al., (Cai et al., 2022) presented a self-supervised learning framework for speaker recognition, which combined clustering and deep representation learning. With multi-modal training data, their framework obtained an accuracy of 77.60% on the VoxCeleb2.

3 Multimodal Learning Strategies

This work uses two single-modality systems to analyze each modality before exploring multimodal learning strategies. For the first system, VoiceNet, we examined two methods. The first one is the x-vector, which is a one-dimensional CNN with five convolutional layers, a statistical pooling layer (Wang et al., 2021), and three fully connected layers that focus on identity identification from voice modality. The architecture of this neural network is depicted in Figure 1. The second one is the gammatonegram, which represents the utterances as an image and fine-tunes the Darknet19 (Redmon and Farhadi, 2017) for learning the features of each speaker and classification task.

As shown in figure 1, the person's speech is extracted from the video file, and silence is removed using a Voice Activity Detector (VAD) (Giannakopoulos, 2009). The audio files are then transformed into MFCC (Mel-frequency cepstral coefficients) features, and the mismatch between training and test utterances is reduced using the CMVN (Cepstral Mean and Variance Normalization) technique (Prasad and Umesh, 2013) and serves as input for VoiceNet. We made another VoiceNet using gammatonegram separately. The final layer of VoiceNet consists of 118 neurons, representing the number of speakers. The outputs of VoiceNets present the performance of the person identification systems based on voice modality.

In addition to VoiceNet, the FaceNet is proposed for face recognition, as shown in Figure 1. FaceNet is built using a pre-trained CNN known as VGGFace2, which has been trained on more than 3.3 million face images (Cao et al., 2018). The transfer learning technique is employed, where the final layers of the network are replaced with new layers to learn the specific information of the new individuals.

The output of FaceNet represents the result of person identification using the face modality. To ensure comparability across different multimodal learning scenarios, an overall audio file and a single frame of the face are extracted from each video file. Since each video contains only one person, the 25th frame is extracted as the face modality.

Figure 1 illustrates the three proposed strategies for multimodal learning: sensor fusion, feature fusion, and score fusion. This study

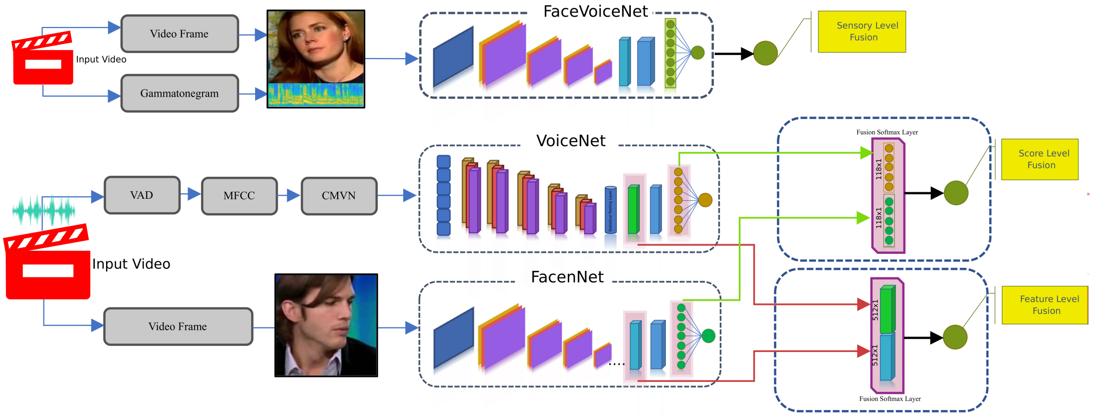


Figure 1: The block diagram of the proposed single-modality systems consists of VoiceNet, FaceNet, and three multimodal systems with different fusion strategies

uses FaceNet and VoiceNets as the foundation for feature and score fusion systems. However, We employ a separate network with mixed modalities as input for the sensor fusion mode. In the subsequent sections, we comprehensively explain each multimodal learning strategy.

3.1 Sensor Level Fusion

Since images are two-dimensional and sounds are one-dimensional signals in this multimodal learning strategy, we aim to combine them into a shared data space. We chose the gammatonegram visualization method (Farhadipour et al., 2014) to represent audio files. Sensor-level multimodal learning is performed on the VGGFace2 network and using transfer learning. As depicted in the sensor level fusion system in Figure 1, we create an integrated image to serve as input for the FaceVoiceNet. This input image incorporates both facial and voice information. The reasons for choosing voice as the modality to be mapped to a new space were influenced by two factors: the availability of a powerful pre-trained network for face recognition and the common practice of representing voice in the form of an image using the gammatonegram.

3.2 Feature Level Fusion

In the feature fusion mode, as depicted in Figure 1, a softmax layer is employed for multimodal learning. In this mode, VoiceNet is utilized to extract the x-vector, resulting in a feature vector of speakers with dimensions of 512x1. Similarly, from the FaceNet, the activation values of the last ReLU layer are extracted as the facial feature embedding, which is a 512x1 vector. These two vectors are concatenated and construct a bimodal vector with a size of 1024x1. In another scenario, we replace gammatonegram and Darknet with the x-vector. In this situation, we extract a 118x1 vector from a middle layer of Darknet19 as a feature vector to concatenate with facial features and create a 630x1 dimension multimodal feature vector.

3.3 Score Level Fusion

The score fusion strategy aims to reach a function that accurately predicts the correct speaker ID by using the score vectors from the basis networks. In other words, our proposed system tries to estimate the f in eq. (1) to make D, which is the ground truth in the training process and the final decision in the testing phase.

$$D = f(x, y) \tag{1}$$

In this equation, x and y are the score vectors of VoiceNets and FaceNet, respectively. We used the softmax layer for modeling f based on its high modeling ability. Each input vector has 118 components, resulting in a bimodal vector with a dimension of 236x1.

4 Evaluation Setup

In this work, the test part of the VoxCeleb2 dataset is used for evaluation. This part consists of 118 speakers and is extracted from YouTube videos in real-world conditions, including various types of noise such as laughter, cross-talk, channel effects, music, and some other environmental sounds. We used this subset just to compare the performance of different strategies proposed in this work in a common situation. The test section of VoxCeleb2 comprises a total of 4,911 unique video files and 36,237 utterances extracted from these videos (Chung et al., 2018). To evaluate the performance of the systems, the K-fold cross-validation technique is employed, with a value of K=3 based on a traditional strategy to use 70% of data for training and the rest for testing. This means that the evaluation is repeated three times, and in each iteration, two folds of the dataset are used for training, while the remaining fold is employed for testing. Under these conditions, approximately 24,000 bimodal utterances are used for training, while around 12,000 bimodal utterances are used for testing.

VoiceNet was trained in 8 epochs, with 128 batch sizes, and the learning rate was adjusted from 1e-3 to 1e-5. On the other hand, the VGGFace2 is trained in 15 epochs with batch size 32 and a constant learning rate of 1e-3. For fine-tuning the Darknet19, we set the learning rate to 1e-4 with 15 epochs. Cross-entropy was used as a loss function in the training process of CNNs and softmax layers. Moreover, stochastic gradient descent with momentum was utilized as the optimizer. The training process is stopped for multimodal learning using the softmax layer when the minimum gradient criteria are satisfied. In this work, it occurred around the 40th to 50th epoch. Two data augmentation techniques were used to avoid overfitting and making a general system: rotation operators within the range of [-20, 20] degrees and vertical and horizontal translation by a distance of [5, 5] pixels.

To evaluate proposed identification systems, there exist several metrics that can be utilized to evaluate the efficiency of a multi-class pattern recognition system. The system's effectiveness can be demonstrated from various perspectives by carefully selecting the appropriate metrics that enable comparisons with other works. The proposed evaluation parameters encompass precision, specificity, sensitivity, accuracy, and F1 score. The utilization of a confusion matrix allows for a visual representation of these parameters (Lever, 2016). Verification systems have encountered two types of errors: false acceptance and false rejection. False acceptance refers to the claims that were accepted inaccurately. However, false rejection deals with the identity that is rejected incorrectly. Equational Error Rate (EER) is the optimum point at which these two errors are equal.

5 Experimental Results

Our main goal is to compare different strategies in modality fusion. In this part, we report experimental results in two scenarios. Person identification results are reported to compare the performance of different fusion modes in multi-class classification scenarios in equal situations. Based on the results of the identification task, the best fusion mode is evaluated in the verification scenario as a two-class classification besides single modality modes.

5.1 Person Identification

According to the results in Table 1, the performance of the proposed person identification systems in the single modality can be observed in face identification and two other speaker identification separated based on the x-vector and gammatonegram representation methodologies. The table includes the achievements for each fold separately and the average performance. Based on the results, it is evident that the system performs better in face single modality recognition than voice. The accuracy achieved by the FaceNet is 96.00%, while the VoiceNet reached an accuracy of 72.67% using x-vector and an accuracy of 61.64% based on gammatonegram. It seems that the x-vector could represent the speaker feature better than the gammatonegram in single modality mode.

Other parameters that provide more insight into the systems' performance can also be found in Table 1. These parameters present the ability of systems to accept correct utterances and reject incorrect utterances for each class. It is worth noting that the presence of babble noise and low sound quality significantly impact the performance of VoiceNets, resulting in its lower accuracy compared to FaceNet.

Metrics	Face Identification				x-Vector Speaker Identification				Gamma. Speaker Identification			
	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.
Precision	0.97	0.97	0.95	**0.96**	0.68	0.77	0.74	**0.73**	0.60	0.64	0.60	**0.61**
Sensitivity	0.97	0.97	0.95	**0.96**	0.68	0.77	0.74	**0.73**	0.60	0.64	0.60	**0.61**
Specificity	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**
F-measure	0.97	0.97	0.95	**0.96**	0.68	0.77	0.74	**0.73**	0.60	0.64	0.60	**0.61**
Accuracy(%)	96.54	96.47	94.99	**96.00**	67.59	76.96	73.47	**72.67**	60.45	64.47	60.00	**61.64**

Table 1: Results of single-modality face identification and speaker identification systems with three different feature sets for three folds and presentation of average values

Metrics	Sensor Fusion				Score Fusion				Feature Fusion			
	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.
Precision	0.95	0.95	0.91	**0.94**	0.97	0.97	0.95	**0.96**	0.99	0.99	0.98	**0.99**
Sensitivity	0.95	0.95	0.91	**0.94**	0.97	0.97	0.95	**0.96**	0.99	0.99	0.98	**0.99**
Specificity	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**
F-measure	0.95	0.95	0.91	**0.94**	0.97	0.97	0.95	**0.96**	0.99	0.99	0.98	**0.99**
Accuracy(%)	95.02	95.10	90.70	**93.61**	96.73	96.81	95.18	**96.24**	98.59	98.94	97.59	**98.37**

Table 2: Performance of proposed multimodal identification systems in three different fusion strategies based on gammatonegram and facial features

Metrics	Score Fusion				Feature Fusion			
	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.
Precision	0.97	0.97	0.96	**0.97**	0.99	0.99	0.97	**0.98**
Sensitivity	0.97	0.97	0.96	**0.97**	0.99	0.99	0.97	**0.98**
Specificity	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**
F-measure	0.97	0.97	0.96	**0.97**	0.99	0.99	0.97	**0.98**
Accuracy(%)	96.88	97.27	95.94	**96.70**	98.81	98.88	97.28	**98.33**

Table 3: Results of proposed multimodal identification systems in two different fusion strategies based on x-Vector and facial features

In addition, it can be seen that the modalities can complement each other while being independent of each other. Table 2 presents the results of three multimodal strategies, each based on different fusion methods. In this table, gammatonegram is used for voice and FaceNet features for face modality. The first part of the table illustrates the performance of the proposed multimodal learning system in sensor fusion mode. Despite achieving precision, sensitivity, specificity, F-measure, and accuracy percentage of 0.94, 0.94, 1, 0.94, and 93.61%, respectively, the multimodal system in sensor fusion mode underperforms compared to the FaceNet single modality system.

This suggests that gammatonegram as a speech presentation method can potentially confuse the sensor-level multimodal system, leading to a decrease in performance in comparison with the single modality system. Although the multimodal learning system with input-level fusion of image and speech data does not improve efficiency, it provides valuable insights to researchers. The findings indicate that the fusion of sensors for these two modalities may not significantly enhance performance, underscoring the need for careful consideration of fusion strategies and modality compatibility in multimodal learning systems.

Based on the investigation in Table 2, the multimodal system in score fusion mode has been evaluated using the information from previous single-modality systems based on facial features and gammatonegram representation. Each network's final softmax layer data is used as input for the aggregated softmax layer in multimodal learning. The average performance across three different folds shows that the system achieves better precision, sensitivity, specificity, F-measure, and accuracy scores than the single modality mode. Specifically, the system achieves 96.24% accuracy, which is 0.24% higher than the result obtained by the FaceNet single-modality system.

Table 2 reveals that the proposed softmax layer in feature fusion mode demonstrates high efficiency, achieving 98.37% accuracy, 0.99 precision, 0.99 sensitivity, 1 specificity, and 0.99 F-measure. These results signify a significant improvement in the accuracy of the person identification task, with a 2.37% increase compared to the performance of FaceNet in single modality mode. This underscores the effectiveness of the feature fusion strategy in enhancing the system's accuracy. The additional source data from different modalities provide extra information for identification, as they express different aspects of the same class. For instance, in fold 1, the speaker identification has the lowest

Metrics	Face Verification				Speaker Verification			
	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.
EER(%)	0.48	0.50	2.07	**1.01**	5.99	5.37	3.65	**5.12**

Table 4: Results of proposed person verification systems in two single modes

Metrics	Gammatonegram and FaceNet				x-Vector and FaceNet			
	Fold 1	Fold 2	Fold 3	Avg.	Fold 1	Fold 2	Fold 3	Avg.
EER(%)	0.44	0.41	1.61	**0.82**	0.37	0.36	1.15	**0.62**

Table 5: Results of proposed person verification systems in feature fusion modes based on two different speech features

performance compared to the other two folds, but the facial recognition for this fold is done with high accuracy. This demonstrates that modality fusion can compensate for the shortcomings of single modalities.

Based on Table 3, the proposed softmax layer for facial features and x-vector in score fusion mode reaches 96.7% accuracy on average, which is better than the single modality mode. However, the feature fusion mode showcases remarkable effectiveness, achieving an accuracy of 98.33%, precision and sensitivity scores of 0.98 each, a specificity of 1, and an F-measure of 0.98. These findings underscore the successful enhancement in accuracy for the person identification task by 2.33% compared to FaceNet as the best achievement in single modality mode.

The average achievements of all five modes show that FaceNet performed the best in single-modality scenarios. Despite VoiceNet's lower efficiency, when combined with the face embedding feature using gammatonegram and x-vector, it improved the identification system's performance in two scenarios: feature fusion and score fusion. Our best achievement was in the feature fusion mode, where we combined facial features with the gammatonegram.

5.2 Person Verification

In this section, we investigate the results of the proposed system for the person verification task. Based on the results obtained in person identification, the feature-level fusion has shown the best performance. Therefore, in the verification scenario, we only evaluate the fusion of features consisting of gammatonegram, x-vector, and FaceNet's activations in multimodal mode, besides two typical single modality-based person verification. In this feature fusion mode, the proposed person verification system utilizes Within-Class Covariance Normalization (WCCN) and Linear Discriminant Analysis (LDA) to reduce intra-class variation and decrease the dimension of the feature vector, respectively. The LDA output yields a vector with 150 components as an eigenvector from the original vector with 1,024 elements.

Furthermore, Gaussian Probabilistic Linear Discriminant Analysis (GPLDA) is employed for decision-making. It is trained with 20 iterations and uses vectors with a length of 150 components. These parameters remain fixed in both single and multimodal scenarios. The results of this evaluation can be found in Table 4.

In the single modality mode in verification tasks, both proposed FaceNet and VoiceNet are used as feature extractors, and their features are fed into GPLDA for decision-making in a two-class classification. The results indicate that the average EER in the speaker verification using voice modality is 5.12%, while in the face verification mode, it is 1.01%. As expected, the system based on face modality performs better due to the superior performance of the proposed face feature extraction. In multimodal mode, we designed two systems based on two different voice features.

In the fusion of x-vector and facial features, the system achieves an average EER of 0.62% in the feature fusion mode. however, in concatenating the gammatonegram feature vector with the facial feature vector, the system reaches 0.82% of EER. Similar to the speaker identification scenario, this demonstrates an improved performance compared to the single modality mode.

6 Discussion

visualization of the activity status of layers can also provide useful information for understanding CNN's functionality. Usually, an appropriate cognition of what happens inside the CNN network gets little attention, and one of the approaches is to consider the CNN as a black box. However, there are helpful visualization methods to understand what is happening inside of a CNN. We depict a representation method called the

Locally-Interpretable Model-agnostic Explanation (LIME) technique (Ribeiro et al., 2016) to show the crucial parts of the image that play a more significant role in network decision-making.

This visualization was made using 2D-CNNs consisting of FaceNet in Figure 2 and gammatonegram-based speaker identifiers in Figure 3. These figures make it possible to understand which parts of the images are focused on by the network. It is also possible to understand whether the network is focused on the essential and discriminative parts of the image or not. The image segmentation colors show each part's priority according to a color bar. By examining the map and its corresponding image, it can be seen that the network focused on the distinguishing parts of the image, which is the reason for the acceptable result for FaceNet and weak performance on speaker identification using the gammatonegram feature.

In this work, we tried to analyze different strategies for audio-visual modality fusion in the identity recognition task. However, it could be helpful to compare the results of the proposed systems with previous work to understand the scale of the metrics. The results of the present work depict that utilizing gammatonegram representation for voice modality and VGGFace2 pre-trained network for face modality can properly depict the identity information in the identification task. However, utilizing x-vector as a voice feature vector beside fine-tuned VGGFace2 in the verification tasks can present the best result.

7 Conclusion

In this work, we analyzed different strategies for audio-visual modality fusion in the identity recognition tasks. The results of the present work depict that utilizing gammatonegram representation for voice modality and VGGFace2 pre-trained network for face modality can properly depict the identity information.

This study developed two separate single-modality systems for voice and face in two identification and verification tasks. The FaceNet architecture utilized VGGFace2, while we had two VoiceNet, the first one was trained from scratch to extract the x-vector, and the second one was based on gammatonegram representation and fine-tuning of pre-trained Darknet19. Based on these systems, multimodal learning was applied in three fusion modes, as described in the article. A softmax layer was the classifier in feature fusion and score fusion approaches in the person identification task. However, for person verification, LDA was used for dimension reduction, and GPLDA was utilized for decision-making. The evaluation was conducted on 118 speakers from the VoxCeleb2 dataset. The results demonstrated that combining speech and face modalities using multimodal learning outperformed the single-mode approach in both identification and verification tasks. Additionally, the feature fusion mode was found to be the most effective strategy for these two modalities.

Future studies could enhance this research by investigating more efficient speech features, such as deep belief networks in autoencoder architecture, to be used as input for the proposed VoiceNet for x-vector extraction. Moreover, utilizing para-linguistic systems, such as gender recognition from face or voice, could offer valuable insights in a score fusion scenario.

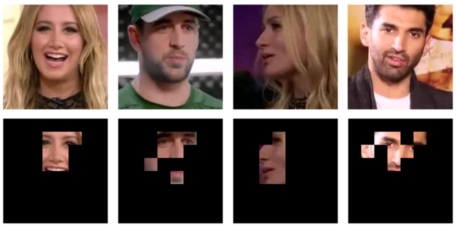


Figure 2: Crucial parts of face images for decision-making in the FaceNet

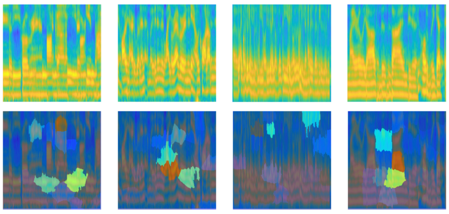


Figure 3: Important part of gammatonegram image from the viewpoint of Darknet19 network for classification

References

Mohammad Rafiqul Alam, Mohammed Bennamoun, Roberto Togneri, and Ferdous Sohel. 2015. A confidence-based late fusion framework for audio-visual biometric identification. *Pattern Recognition Letters*, 52:65–71.

Boucetta Aldjia and Boussaad Leila. 2021. Sensor level fusion for multi-modal biometric identification using deep learning. In *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–5.

Vahid Asadpour, Mohammad Mehdi Homayounpour, and Farzad Towhidkhah. 2011. Audio–visual speaker identification using dynamic facial movements and utterance phonetic content. *Applied Soft Computing*, 11(2):2083–2093.

Danwei Cai, Weiqing Wang, and Ming Li. 2022. Incorporating visual information in audio based self-supervised speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1422–1435.

Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. 2018. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*.

Aref Farhadipour, Mohammad Asgari, and Mohammad Reza Hasanabadi. 2014. Gammatonegram based speaker identification. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 52–55. IEEE.

Aref Farhadipour and Pouya Taghipour. 2023. Facial emotion recognition under mask coverage using a data augmentation technique. In *2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 001–006. IEEE.

Aref Farhadipour and Hadi Veisi. 2024a. Analysis of deep generative model impact on feature extraction and dimension reduction for short utterance text-independent speaker verification. *Circuits, Systems, and Signal Processing*, pages 1–18.

Aref Farhadipour and Hadi Veisi. 2024b. Gammatonegram representation for end-to-end dysarthric speech processing tasks: speech recognition, speaker identification, and intelligibility assessment. *Iran Journal of Computer Science*, pages 1–14.

Theodoros Giannakopoulos. 2009. A method for silence removal and segmentation of speech signals, implemented in matlab. *University of Athens, Athens*, 2.

Jaesung Huh, Andrew Brown, Jee weon Jung, Joon Son Chung, Arsha Nagrani, Daniel Garcia-Romero, and Andrew Zisserman. 2023. Voxsrc 2022: The fourth voxceleb speaker recognition challenge. *arXiv preprint arXiv:2302.10248*.

Jake Lever. 2016. Classification evaluation: It is important to understand both what a classification metric expresses and what it hides. *Nature methods*, 13(8):603–605.

Shervin Minaee, Amirali Abdolrashidi, Hang Su, Mohammed Bennamoun, and David Zhang. 2023. Biometrics recognition using deep learning: A survey. *Artificial Intelligence Review*, pages 1–49.

Abderrazzaq Moufidi, David Rousseau, and Pejman Rasti. 2023. Attention-based fusion of ultrashort voice utterances and depth videos for multimodal person identification. *Sensors*, 23(13):5890.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2020. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027.

Catherine Perrodin, Christoph Kayser, Taylor J Abel, Nikos K Logothetis, and Christopher I Petkov. 2015. Who is that? brain networks and mechanisms for identifying individuals. *Trends in cognitive sciences*, 19(12):783–796.

Ashwin Prakash, Athira Nambiar, Alexandre Bernardino, and et al. 2023. Multimodal adaptive fusion of face and gait features using keyless attention based deep neural networks for human identification. *arXiv preprint arXiv:2303.13814*.

N Vishnu Prasad and Srinivasan Umesh. 2013. Improved cepstral mean and variance normalization using bayesian framework. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 156–161.

Yanmin Qian, Zhengyang Chen, and Shuai Wang. 2021. Audio-visual deep neural network for robust person verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1079–1092.

N. Venkata Ramana, S. Anu H Nair, and KP Sanal Kumar. 2022. Hybrid biometric based person identification using machine learning. *Journal of Pharmaceutical Negative Results*, pages 1605–1614.

Joseph Redmon and Ali Farhadi. 2017. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Leda Sarı, Kritika Singh, Jiatong Zhou, Lorenzo Torresani, Nayan Singhal, and Yatharth Saraf. 2021. A multi-view approach to audio-visual speaker verification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198.

Saqlain Hussain Shah, Muhammad Saad Saeed, Shah Nawaz, and Muhammad Haroon Yousaf. 2023. Speaker recognition in realistic scenario using multimodal data. In *2023 3rd International Conference on Artificial Intelligence (ICAI)*, pages 209–213.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333.

Anton Stefanidi, Artem Topnikov, Gennadiy Tupitsin, and Andrey Priorov. 2020. Application of convolutional neural networks for multimodal identification task. In *2020 26th Conference of Open Innovations Association (FRUCT)*, pages 423–428.

Ruijie Tao, Rohan Kumar Das, and Haizhou Li. 2020. Audio-visual speaker recognition with a cross-modal discriminative network. *arXiv preprint arXiv:2008.03894*.

Rui Wang, Junyi Ao, Long Zhou, Shujie Liu, Zhihua Wei, Tom Ko, Qing Li, and Yu Zhang. 2022. Multi-view self-attention based transformer for speaker recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6732–6736.

Shuai Wang, Yexin Yang, Yanmin Qian, and Kai Yu. 2021. Revisiting the statistics pooling layer in deep speaker embedding learning. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5.

PoliTun: Tunisian Political Dataset for Detecting Public Opinions and Categories Orientation

Chayma Fourati
National School of Comupter
Science / Tunisia
chaymafourati12@gmail.com

Roua Hammami
Higher Institute of Multimedia Arts / Tunisia
roua.hammami@yahoo.com

Chiraz Latiri
Faculty of Sciences of Tunis,
Tunis EL Manar University / Tunisia
chiraz.latiri3@gmail.com

Hatem Haddad
Manouba University / Tunisia
haddad.hatem@gmail.com

Abstract

In recent years, social media platforms and online forums have become essential sources of political discourse, reflecting the ever-evolving dynamics of societal opinions and sentiments. With the freedom of expression privilege granted after the Tunisian revolution, sensitive "taboo" topics such as political issues have become popular and widely discussed by Tunisians across social media platforms. However, on the downside, it has become easy to spread abusive/hate propaganda against individuals or groups. To address this gap, we introduce PoliTun, a new dataset designed specifically for political analysis in the Tunisian dialect, aiming to illuminate Tunisia's political landscape from a linguistic perspective. We describe the methodology used for constructing the PoliTun dataset, including data collection, preprocessing, and annotation. Then, we present experiments conducted with PoliTun for category detection and political opinion identification, utilizing various machine learning, deep learning and transformer-based models. The results reveal variable model performances. In conclusion, the development of PoliTun represents a significant advancement in political analysis in Tunisia, providing a foundation for nuanced exploration of political discourse in this unique sociopolitical landscape. PoliTun will be available upon request[1].

1 Introduction

Tunisia represents a country known for its vibrant political scene and diverse linguistic landscape, where the Tunisian Dialect represents its key form of expression. With the rise of social media platforms and the increasing prevalence of online discussions, the study of political discourse has gained prominence among researchers and policymakers. Particularly, the explosion of user-generated content on Twitter provides a valuable resource for mining insights on diverse subjects, including political content. On July, 25th, 2021, political decisions created a new political environment in Tunisia leading to various opinions across citizens. In fact, On July 25, 2021, the President of the Republic, Kaïs Saied, applied Article 80, dismissing the Head of Government, suspending all the activities of the National Assembly, and seizing full powers.

This situation split Tunisians into groups: loyalists to ideas with uncritical support, hunters for opportunities around, skeptics about choices, and opponents of slides towards an undemocratic political system. This cleavage caused intense debates between Tunisians, most of which took place on social media networks, leading to a high emergence of abusive/hate speech, polarization, and conspiracism. Hence, the need to create automatic solutions to detect such behaviours. However, the majority of existing studies tend to overlook the importance of regional dialects, thus limiting our understanding of political dynamics in specific linguistic contexts. The Tunisian Dialect, a variety of Arabic influenced by Berber, French, and other languages, serves as a distinct mode of communication among Tunisians, particularly in informal settings. It has unique linguistic features, idioms, and expressions that reflect the local cultural, historical, and political nuances. Therefore, an analysis of political discourse solely based on the Modern Standard Arabic or major languages would fail to capture the subtleties and intricacies embedded within the Tunisian Dialect. To address this research gap, we present the PoliTun dataset, a comprehensive collection of political texts in the Tunisian Dialect

---

[1] Please contact the authors via email to obtain the link.

sourced from Twitter. The dataset includes a wide range of political topics, covering elections, governance, public policies, activism, and more, to provide a holistic view of the political landscape in Tunisia. By focusing on the Tunisian Dialect, PoliTun enables researchers and language enthusiasts to go into the characteristics of political discourse within the unique socio-cultural context of Tunisia. In this paper, we provide an overview of related work in the field of political analysis and dialectal studies. Then, we describe the methodology employed in constructing the PoliTun data where we outline the methodology employed in compiling the PoliTun dataset, including data collection, preprocessing, and annotation procedure. Then, we present experiments performed using our dataset for category detection and political opinion detection. Finally, we present the conclusion and future work.

2 Related works

According to the Larousse online dictionary[2], an opinion can be defined as a judgment or feeling expressed by an individual or group on a subject or facts, reflecting their perception and thoughts, or a formal statement giving the reasons behind a given judgment. Opinion detection has become one of the most active fields of study in natural language processing since the early 2000s (Liu et al., 2010); (Liu, 2022). (Pang et al., 2008) conducted an in-depth study covering various aspects of opinion analysis, such as opinion extraction, sentiment classification, polarity analysis and opinion synthesis. This analysis included different approaches, whether lexicon-based or supervised learning, using similarity measures and various classifiers. Other work in the literature has compared several approaches, including naive bayes classifiers, support vector machines (SVMs) (Palau and Moens, 2009), logistic regression (Levy et al., 2014) . Applying the CNN deep learning approach in a mobile environment a study (Kalaivani and Jayalakshmi, 2021) proposed sentiment analysis based on movie reviews. Their approach involved the use of Polarity, IMDb and Rotten Tomato datasets, Results indicate that the integration of GloVe word vectors led to better performance. Recurrent and recursive neural networks were examined with different types of Arabic-specific processing (Al Sallab et al., 2015); (Al-Sallab et al., 2017); (Baly et al., 2017). Convolutional neural networks (CNNs) were trained using pre-trained word embeddings (Dahou et al., 2019). A hybrid model was proposed by (Farha and Magdy, 2019), where CNNs were used for feature extraction and LSTMs were used for sequence and context understanding. There is a lack of pre-trained language models, which limits the performance of NLP applications for some languages. The article (Kenton and Toutanova, 2019) highlights the ineffectiveness of traditional models as they require task-specific datasets, making them impractical to use. Unlike English, Arabic has a rich morphology and limited resources, as it has many dialects, which makes Automatic Comprehension of this language complex due to linguistic variations. There is a need to evaluate these models consistently and on various NLP tasks in Arabic. The article (Abdul-Mageed et al., 2020) presents the creation of two powerful language models specific to MarBERT and MarBERT-v2, pre-trained on massive and diverse datasets, including datasets, including social media data. Faced with these challenges, the researchers (Antoun et al., 2020) set out to create a solution capable of efficiently processing the Arabic language, it became imperative to develop a specific language processing model. Although pre-trained language models such as BERT have proven effective in English, their direct application to Arabic proved less conclusive. The morphological complexity of the Arabic language requires careful adaptation to exploit the full potential of these models. Hence, the idea of creating AraBERT emerged with the aim of meeting the unique challenges of the Arabic language and providing a powerful tool for the Arabic NLP community.

In (Abd et al., 2020), the authors present a Political Arabic Articles Data Set entitled PAAD, comprising 206 articles classified into Reform, Conservative, and Revolutionary collected from newspapers, social networks, general forums, and ideological websites. This data set is oriented at Arabic Computational Linguistics by providing a valuable resource for political text classification in Modern Standard Arabic and includes only three labels.

Despite this advancement, it does not cover the aspect of dialectal variations, which continues to be a significant challenge in Arabic NLP. To the best of our knowledge, our work presents the first dataset dedicated for political text in an Arabic dialect, particularly the Tunisian one.

---

[2] https://www.larousse.fr/

3 PoliTun Dataset: Data Building From Twitter

In this section, we detail the creation and the particularities of PoliTun, our large political opinion and category detection dataset composed of about 30K tweets collected about events after July 25th, 2021 Tunisian events.

The majority of the sentences deal with the post-July 25 period and the agitation of Tunisians on social networks in relation to the current political dynamic. In this work, we aim to offer a deep understanding of the diverse themes within political discussions, particularly, in the Tunisian context.

3.1 Data Collection

We collected tweets using the Twitter streaming API. We have collected and scraped tweets using Twitter hashtags related to the Tunisian Political Context. We manually extracted a list of more than **200 hashtags** that were used for scrapping tweets. Examples are the following: #لا_للمحاكمات_العسكرية, #تفكيك_منظومة_الفساد_السياسي, #تونس_جمهورية_مدنية_ديمقراطية, etc.

In order to make sure that all tweets are after July 25th event, a python script was created to extract only tweets subsequent to that date. Another python script was created to split data into Arabic, Tunizi which is the Tunisian Dialect written in Latin letters, French and English sentences. Examples from the dataset and their translation are presented in Table 1.

3.2 Data Preprocessing

The collected tweets were not clean as they included many punctuation marks including, hashtags, Emojis, and more. Therefore, the following steps have been followed to ensure the quality of this dataset:

- **Removing Punctuation:**

  The tweets in the dataset contained a mix of Arabic and non-Arabic punctuation. in Tunisia, the use of punctuation in written texts is relatively low compared to other languages. Hence, these marks are generally outliers and do not contribute to the overall understanding of the text. Hence, different punctuation marks were removed using a python script.

- **Removing Emojis:** In Tunisia, the use of sarcasm is quite often, both in real life and over social networks which, generally, leads to carrying different meanings for the same emoji which can lead to potential confusion for the learning models. Since there are no available libraries that can translate emojis into Tunisian, we removed emojis present in our dataset.

- **Removing duplicates and retweets:** Duplicate data can skew the results of the analysis of results. Since the model will give it undue weight, it would lead to biased or inaccurate predictions.

- **Keeping only Arabic-letters tweets for annotation:** We kept only the tweets in Arabic letters for annotation. We leave including Latin letters as future work.

Finally, we sampled about 30,000 tweets written in Arabic letters for annotation in total.

3.3 Data Annotation

The labels of the dataset were identified by three sociologists, experts in their domain after reading multiple randomly selected examples of the scrapped tweets.

The dataset was annotated by three Tunisian female native speakers, that are involved in civil society, all aged 25. Due to the limited number of annotators, only 30k data in Arabic letters was annotated divided equally on the annotators.

To ensure the quality of the annotaion, we take 100 internally annotated examples and ask for review from the sociologists. If an annotator mislabels more than 25% of these examples, we discard the annotations and ask her to relabel them based on the comments of the sociologists.

In our dataset, each sentence is labeled twice, considering two distinct aspects: **category** and **opinion**. The category aspect includes six labels, while the opinion aspect consists of three labels.

3.3.1 Category Identification

Regarding the category label, Six categories were identified:

- **Hate Speech**: A tweet that discriminates, stigmatizes, or incites violence or prejudice against individuals or groups based on attributes such as race, ethnicity, religion, gender, sexual orientation, disability, or other characteristics.

Table 1: Political Data Examples.

Tweet	English Translation
لله يبارك تونس باهية وانت سبب بهاها	God bless you. Tunisia is nice and you're the reason for it.
تونس الجزائر بلد حدودي معانا و بينا مصالح	Tunisia, Algeria is a border country with us and we have interests together.

- **Polarization**: mainly expresses the position of being for or against a political position. It often conveys a strong and uncompromising stance, contributing to an "us versus them" mentality. Such tweets may use language that reinforces the separation between conflicting perspectives and may lack nuance or a willingness to engage in constructive dialogue.

- **Conspiracy**: expressed more in the words of a political person or his/her supporters. It expresses a belief or suspicion about a secretive or covert plan that is allegedly being carried out by powerful individuals or organizations.

- **Denunciation**: is the recounting of a political event without taking a position. It refers to the act of publicly condemning or expressing disapproval of someone or something, often due to perceived wrongdoing, unethical behavior, or actions contrary to societal norms or values. It involves making an official or public statement declaring strong disapproval or condemnation.

- **Skepticism**: an attitude of doubt or disbelief towards claims, beliefs, or assertions, particularly those that are commonly accepted or taken for granted. Skepticism can manifest in various forms, ranging from questioning the validity of specific claims to adopting a general stance of skepticism towards all knowledge claims until sufficient evidence is provided.

- **Off-Topic**: the case where the tweet doesn't deal with any political discourse or subject relating to the Tunisian context.

3.3.2 Opinion Identification

The same tweet is also annotated with one of the three opinion labels:

- **Positive**: expresses a positive opinion regarding an event/idea.

- **Negative**: expresses a negative opinion regarding an event/idea.

- **Neutral**: without interest/perspective in the subject.

Tweets that are annotated as off-topic do not have an opinion label.

Examples from the dataset with their annotations are presented in Table 2.

3.4 Data Statistics

Statistics of the initial data after preprocessing and cleaning are presented in Table 3.

The annotated data includes about 30k tweets written in Arabic letters.

Table 4 and Figure 1 show the distribution of the annotated tweets by category (Off-topic, Denunciation, Polarization, Conspiracy, Skepticism, Hate speech). We note a clear superiority in number of comments categorized as “Off Topic”, with a total of 12,503 annotations representing 42% of the data, compared to the other categories. The “Denunciation” and “Polarization” categories come next, with 7,763 and 7,660 annotated comments representing 26% and 25% respectively. “Conspiracy” includes 1,072 annotations, “Skepticism” with 528 annotations, and finally “Hate speech” with 448 annotations. The three last labels present only 7% of the dataset.

Figure 2 summarizes the distribution of the annotated tweets according to their opinion polarity (Positive, Negative, Neutral). We note that Negative comments outnumber Positive ones by a large margin: 57% were annotated as negative, 32% as positive, and 11% as neutral.

4 Experimental Setup

We divide our data into 80% for training and 20% for testing. We run experiments using the following machine learning and deep learning models: Naive Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR), Convolutional

Table 2: Political Data Examples with their annotation.

Tweet	English Translation	Category	Opinion
ياخي وينهم مواطنون ضد الإنقلاب أكاهوا خرجو مرتين ولاثلاثة	where are the Citizens that are against the Coup, they've been out two or three times	Polarization	Negative
والا ترهدين الطبوبي	The hypocrisy of Taboubi	Hate Speech	Negative

Language	#Sentences
Arabic	113341
Tunizi	3685
French	270
English	657
Total	**117953**

Table 3: Initial Data statistics.

Category	#Sentences
Off-topic	12503
Denunciation	7763
Polarization	7660
Conspiracy	1072
Skepticism	528
Hateful speech	448
Total	**29974**

Table 4: Distribution of Annotated Comments by Category.

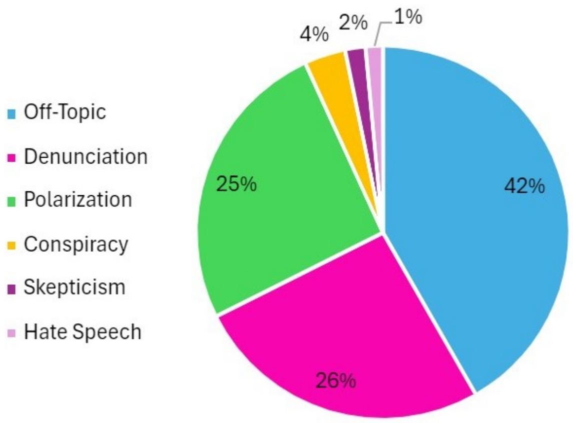


Figure 1: Category Label Percentages.

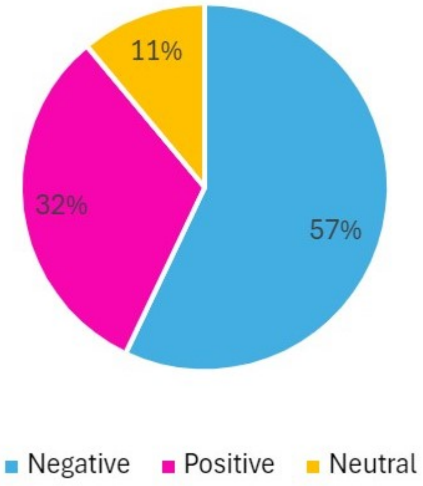


Figure 2: Opinion Label Percentages.

Neural Networks (CNN), and Long-Short Term Memory (LSTM).

When training deep learning models, we use as embedding FastText pretrained on Arabic Wikipedia, batch size equal to 128, and 20 epochs.

Also, different pre-trained models were used in order to achieve the best results. Bacause there is a lack of pretained models for the Tunisian dialect, we chose to experiment with the following models that are pretrained on Arabic language and Arabic dialects respectively:

- **AraBERT** (Antoun et al., 2020): is a BERT based model for Modern Standard Arabic Language understanding, trained on 70M sentences from several public Arabic datasets and news websites.
- **MarBERT** (Abdul-Mageed et al., 2020): is a large-scale pretrained language model using the BERT base's architecture. MARBERT is trained on on 128 GB of tweets from various Arabic dialects containing at least 3 Arabic words. With very light preprocessing the tweets were almost kept at their initial state to retain a faithful representation of the naturally occurring text.

We finetuned BERT models with the following hyperparameters: batch size equal to 128, 20 epochs, and max seq length of 128.

The metrics used to evaluate the model's predictions are accuracy, recall, precision and F1 score (micro and macro).

5 Results and Discussion

In this section, we present the set of experiments performed on both tasks and discuss the results obtained.

5.1 Opinion Classification subtask

Table 5 presents results of the category classification task with Kmeans, Logistic Regression, Naive Bayes and Support Vector Machines.

The results of the evaluation of machine learning models reveal varying metrics analyzed. The best result was acheived using SVM performing 70% accuracy. However, F1 macro score achieved 57% because the dataset is not balanced.

Model	LR	NB	SVM
Accuracy	0.69	0.68	**0.70**
F1 micro	0.69	0.68	0.70
F1 macro	0.58	0.42	0.57
Recall	0.62	0.68	0.55
Precision	0.57	0.76	0.67

Table 5: Performance Metrics of Machine Learning Models for Opinion Classification.

Table 6 presents the results of the deep learning algorithms for the opinion classification task. The CNN outperforms the LSTM model reaching an accuracy value of 95% and F1 macro of 91%.

Model	CNN	LSTM
Accuracy	**0.95**	0.65
F1 micro	0.92	0.68
F1 macro	0.91	0.57
Recall	0.91	0.53
Precision	0.95	0.62

Table 6: Performance Metrics of Deep Learning Models for Opinion Classification.

Table 7 presents results for finetuning the AraBERT and MarBERT models on the Opinion Classification task. In fact, AraBERT achieves an accuracy of 75.42%. while MarBERT outpeforms it acheiving 76% of accuracy measure and 63% F1 macro. This is mainly because our dataset is written in the Tunisian dialect and MarBERT was trained on different dialectal Arabic texts while AraBERT was trained on Modern Standard Arabic (MSA).

Model	AraBERT	MarBERT
Accuracy	0.75	**0.76**
F1 micro	0.64	0.65
F1 macro	0.62	0.63
Recall	0.62	0.63
Precision	0.67	0.66

Table 7: Performance Metrics of BERT variant Models for Opinion Classification.

5.2 Category Classification subtask

Table 8 presents results of the category classification task with Kmeans, Logistic Regression, Naive Bayes, and Support Vector Machines.

Model	Kmeans	LR	NB	SVM
Accuracy	0.45	0.64	0.60	**0.65**
F1-micro	0.45	0.64	0.60	0.65
F1-macro	0.24	0.37	0.21	0.33
Recall	0.30	0.34	0.21	0.31
Precision	0.34	0.45	0.54	0.64

Table 8: Performance Metrics of Machine Learning Models for Category Classification.

In this task, SVM also outperforms the other machine learning models achieving 65% accuracy and 33% F1 macro. In this subtask, F1 macro gives low results because we have a non balanced dataset.

Table 9 presents results of the category classification task with Convolutional Neural Networks and Long Short Term Memory.

Model	CNN	LSTM
Accuracy	**0.70**	0.54
F1 micro	0.96	0.59
F1 macro	0.35	0.33
Recall	0.35	0.29
Precision	0.36	0.64

Table 9: Performance Metrics of Deep Learning Models for Category Classification.

The CNN outperforms LSTM by achieving 70% and 35% accuracy and F1 macro results respectively. Again, due to the imbalance labels in the dataset, F1 macro achieves low results.

Table 10 presents results of finetuning AraBERT and MarBERT models for category identification task.

In this case, MarBERT outperforms AraBERT in terms of accuracy by 3%. However, AraBERT outperforms MarBERT it terms of F1 macro by

Model	AraBERT	MarBERT
Accuracy	0.66	**0.69**
F1 micro	0.32	0.35
F1 macro	0.39	0.32
Recall	0.30	0.32
Precision	0.34	0.37

Table 10: Performance Metrics of BERT variant Models for Category Classification.

7%. But, still achieving low results because we are dealing with an imbalance in the dataset labels.

6 Conclusion and Future work

The development of PoliTun dataset represents a significant step forward in political analysis, particularly within the context of the Tunisian dialect. By creating a dataset of about 30,000 manually annotated data by a team work of both Tunisian sociologists and engaged citizens, we tackled the science of political discourse in Tunisia. Our experiments have showcased the potential of computational methods in understanding complex linguistic dynamics within this unique sociopolitical landscape. Moving forward, several avenues for future research present themselves. Firstly, expanding the PoliTun dataset to include larger range of topics and dialectical nuances which would enhance its utility for comprehensive political analysis. Also, including collaboration between researchers, policymakers, and local communities in Tunisia to co-create and utilize PoliTun for informed decision-making and civic engagement initiatives would be instrumental in maximizing its societal impact. Overall, the continued development and utilization of PoliTun stand to enrich our understanding of political dynamics in Tunisia and beyond, contributing to more inclusive and data-driven approaches to governance and social change.

References

Dhafar Hamed Abd, Ahmed T Sadiq, and Ayad R Abbas. 2020. Paad: Political arabic articles dataset for automatic text categorization. *Iraqi Journal for Computers and Informatics*, 46(1):1–11.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Ahmad Al-Sallab, Ramy Baly, Hazem Hajj, Khaled Bashir Shaban, Wassim El-Hajj, and Gilbert Badaro. 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–20.

Ahmad Al Sallab, Hazem Hajj, Gilbert Badaro, Ramy Baly, Wassim El-Hajj, and Khaled Shaban. 2015. Deep learning models for sentiment analysis in arabic. In *Proceedings of the second workshop on Arabic natural language processing*, pages 9–17.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Ramy Baly, Hazem Hajj, Nizar Habash, Khaled Bashir Shaban, and Wassim El-Hajj. 2017. A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–21.

Abdelghani Dahou, Mohamed Abd Elaziz, Junwei Zhou, and Shengwu Xiong. 2019. Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Computational intelligence and neuroscience*, 2019.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online arabic sentiment analyser. In *Proceedings of the fourth arabic natural language processing workshop*, pages 192–198.

MS Kalaivani and S Jayalakshmi. 2021. Sentiment analysis on micro-blog data using machine learning techniques-a review. In *IOP Conference Series: Materials Science and Engineering*, volume 1049, page 012012. IOP Publishing.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.

Bing Liu. 2022. *Sentiment analysis and opinion mining*. Springer Nature.

Bing Liu et al. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2(2010):627–666.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

Efficient Few-shot Learning for Multi-label Classification of Scientific Documents with Many Classes

Tim Schopf, Alexander Blatzheim, Nektarios Machner, and Florian Matthes
Technical University of Munich, Department of Computer Science, Germany
{tim.schopf,alexander.blatzheim,nektarios.machner,matthes}
@tum.de

Abstract

Scientific document classification is a critical task and often involves many classes. However, collecting human-labeled data for many classes is expensive and usually leads to label-scarce scenarios. Moreover, recent work has shown that sentence embedding model fine-tuning for few-shot classification is efficient, robust, and effective. In this work, we propose FusionSent (**Fusion**-based **Sent**ence Embedding Fine-tuning), an efficient and prompt-free approach for few-shot classification of scientific documents with many classes. FusionSent uses available training examples and their respective label texts to contrastively fine-tune two different sentence embedding models. Afterward, the parameters of both fine-tuned models are fused to combine the complementary knowledge from the separate fine-tuning steps into a single model. Finally, the resulting sentence embedding model is frozen to embed the training instances, which are then used as input features to train a classification head. Our experiments show that FusionSent significantly outperforms strong baselines by an average of 6.0 F_1 points across multiple scientific document classification datasets. In addition, we introduce a new dataset for multi-label classification of scientific documents, which contains 203,961 scientific articles and 130 classes from the arXiv category taxonomy. Code and data are available at https://github.com/sebischair/FusionSent.

1 Introduction

Scientific literature has grown exponentially over the last few decades, with countless new publications being added every year (Dong et al., 2017). To be searchable and accessible to researchers, policymakers, and the public, scientific literature must be managed and categorized in digital libraries (Toney and Dunham, 2022). However, this poses a significant challenge due to the huge volume of documents and the variety of topics they cover (Sadat and Caragea, 2022). In addition to the broad spectrum of possible topics, scientific documents often cannot be assigned to just one topic due to their interdisciplinary character. Consequently, automatically categorizing scientific documents must be approached as a multi-label classification problem over large label spaces. Previous works approach this task either in an unsupervised (Shen et al., 2018; Salatino et al., 2019; Mustafa et al., 2021; Toney and Dunham, 2022; Schopf and Matthes, 2024) or in a fully supervised (Gialitsis et al., 2022; Sadat and Caragea, 2022; E. Mendoza et al., 2022; Schopf et al., 2023) manner. While supervised approaches offer high prediction quality, they require a large corpus of annotated data to perform. Often, however, a large corpus of annotated data is unavailable, e.g., when a new categorization scheme is being developed for an emerging scientific field. Unsupervised approaches provide a possible circumvention of this limitation but are accompanied at the expense of prediction quality.

To improve classification performance in scenarios where labeled data is unavailable, domain experts may annotate a small part of the dataset. However, annotating many classes naturally leads to data scarcity, as collecting sufficient training data for all classes causes significantly higher costs (Xu et al., 2023a). Therefore, to support the classification of scientific documents in such scenarios, we consider the multi-label classification of scientific documents as a few-shot task. Few-shot approaches are designed to train an effective model with a few labeled examples, reducing the cost of developing models for new domains and tasks (Huang et al., 2023).

In recent work, SetFit (Tunstall et al., 2022) demonstrated strong few-shot classification performance by contrastively fine-tuning (Koch et al., 2015) sentence embedding models. Since this approach does not require prompts and is effective on relatively small models, it is much more efficient

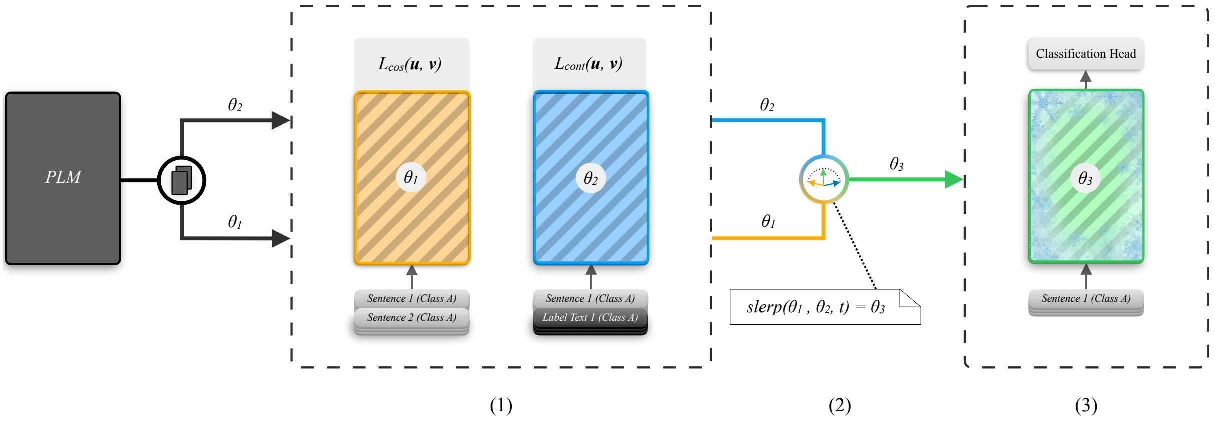


Figure 1: The training process of FusionSent comprises three steps: (1) Fine-tune two different sentence embedding models from the same Pre-trained Language Model (PLM), with parameters θ_1, θ_2 respectively. θ_1 is fine-tuned on pairs of training sentences using cosine similarity loss and θ_2 is fine-tuned on pairs of training sentences and their corresponding label texts, using contrastive loss. Label texts can consist of simple label/class names or of more extensive texts that semantically describe the meaning of a label/class. (2) Merge parameter sets θ_1, θ_2 into θ_3 using Spherical Linear Interpolation (SLERP). (3) Freeze θ_3 to embed the training sentences, which are then used as input features to train a classification head.

and consistent than common prompt-based methods such as In-Context Learning (ICL) (Brown et al., 2020) and Pattern-Exploiting Training (PET) (Schick and Schütze, 2021), which involve careful prompt engineering and large-scale model sizes.

In this paper, we propose *FusionSent*, which builds on the idea of contrastive sentence embedding training for efficient few-shot classification. As illustrated in Figure 1, FusionSent uses the few annotated examples, as well as label texts, to contrastively fine-tune two separate sentence embedding models from the same Pre-trained Language Model (PLM) checkpoint. One model is fine-tuned to maximize similarities between training examples sharing the same class, and the other model is fine-tuned to maximize similarities between training examples and their corresponding label texts. After fine-tuning, the weights of both models are merged to obtain the model body of FusionSent. For subsequent classifier training, the model body is frozen to embed the few training examples, which are then used as input features to train a simple logistic regression head. This approach works effectively with relatively small model sizes, requires no prompts, and merging fine-tuned sentence embedding models incurs no additional inference or memory costs (Wortsman et al., 2022). Our experiments show that FusionSent consistently outperforms various baselines on different datasets for multi-label classification of scientific documents with many classes. Furthermore, we show that FusionSent can improve few-shot performance in multi-class settings of different domains with a small number of classes.

In addition to FusionSent, we introduce a new dataset for multi-label classification of scientific documents. The dataset consists of 130 classes and 203,961 scientific articles that have been manually categorized by their authors into one or more topics from the arXiv category taxonomy[1].

2 Related Work

2.1 Classification of Scientific Documents

Unsupervised approaches typically use embeddings of topics as well as scientific documents and perform classification based on their similarities (Shen et al., 2018; Salatino et al., 2019; Mustafa et al., 2021; Toney and Dunham, 2022). More recently, classifying scientific documents has been regarded as a fully supervised task. SciNoBo (Gialitsis et al., 2022) uses the structural properties of publications and their citations and references organized in a multilayer graph network for predicting topics of scientific publications. HR-SciBERT (Sadat and Caragea, 2022) uses a multi-task learning approach for topic classification with keyword labeling as an auxiliary task. Finally, E. Mendoza et al. (2022) use ensemble models to classify scientific documents into multiple research themes.

---

[1] https://arxiv.org/category_taxonomy

2.2 Few-shot Classification

Prominent techniques for few-shot classification involve ICL, utilizing task-specific prompts with a few labeled examples (Brown et al., 2020). However, while avoiding gradient updates, ICL necessitates large model sizes for good performance, resulting in computationally expensive inference. Conversely, prompt-based fine-tuning proves to be effective with smaller models (Schick and Schütze, 2021; Tam et al., 2021; Gao et al., 2021). Additionally, Parameter Efficient Fine-Tuning (PEFT) can further reduce training costs by fine-tuning a considerably smaller module within a frozen PLM (Houlsby et al., 2019; Li and Liang, 2021; Hu et al., 2022; Karimi Mahabadi et al., 2022; He et al., 2022; Liu et al., 2022; Aly et al., 2023). In contrast to these methods, fine-tuning few-shot classification models via contrastive sentence embedding training provides two primary advantages: (1) it requires significantly smaller model sizes, and (2) eliminates the necessity for prompts or instructions (Tunstall et al., 2022; Huang et al., 2023; Bates and Gurevych, 2024), which can cause significant performance variance and require careful design (Perez et al., 2021).

2.3 Model Fusion

Model fusion, which involves the integration of capabilities from different models, can be mainly divided into two categories. Firstly, ensemble approaches combine the output of multiple models to enhance the overall prediction performance (Littlestone and Warmuth, 1994; Sagi and Rokach, 2018). Outputs are typically combined by weight averaging (Littlestone and Warmuth, 1994) or majority voting (Monteith et al., 2011). These ensemble approaches can improve the prediction performance of large-scale language models (Jiang et al., 2023). Secondly, weight merging approaches enable model fusion at the parameter level. Wortsman et al. (2022) show that weight averaging of multiple models fine-tuned with different hyperparameters improves prediction accuracy and robustness. Task vectors derived from model weights can be modified and combined together through arithmetic operations to steer the behavior of a resulting model (Ilharco et al., 2023). This approach can be enhanced by trimming task vectors and resolving sign conflicts before merging them (Yadav et al., 2023). In addition, Drop And Rescale (DARE) can be used as a general preprocessing technique for existing model merging methods to merge multiple task-specific fine-tuned models into a single model with diverse abilities (Yu et al., 2023).

2.4 Datasets for Topic Classification of Scientific Documents

Various datasets for multi-label topic classification of scientific documents have been introduced. The Cora dataset (McCallum et al., 2000) contains about 50,000 computer science research papers categorized into 79 topics. Several datasets have been released based on the ACM Computing Classification System[2] (Santos and Rodrigues, 2009; Sadat and Caragea, 2022). Schopf et al. (2023) introduce a dataset of 179,349 scientific papers categorized into 82 different NLP-related topics. Yang et al. (2018) create a dataset of 55,840 arXiv[3] papers, in which each paper is assigned to several classes covering 54 different topics. However, this dataset is not publicly available.

3 Background

Sentence Embedding Model Fine-tuning for Few-shot Classification Tunstall et al. (2022) show that sentence embedding models can be used in a two-step training process for efficient few-shot classification. In the first step, a sentence embedding model is fine-tuned in a contrastive manner by sampling positive and negative sentence pairs from few-shot labeled examples. In the second step, the fine-tuned sentence embedding model is frozen to encode all available few-shot examples. The resulting embeddings are then used as input features to train a simple logistic regression classifier (Cox, 1958) as the model head.

Label Texts for Document Classification in Label-scarce Scenarios Xu et al. (2023a) show that mapping representation spaces of training instances to their respective label descriptions in embedding space can be effective in label-scarce classification scenarios. They reformulate classification with many classes as a dense retrieval task and train a dual encoder that learns to maximize the similarity between embeddings of the training instances and their respective label descriptions. During inference, they use the top-k retrieved labels of each instance for classification. Similarly, WanDeR (Xu et al., 2023b) and FastFit (Yehudai and Bendel, 2024) use label names and dense retrieval for

---

[2] https://dl.acm.org/ccs

[3] https://arxiv.org

multi-class classification. However, dense retrieval approaches are challenging to apply in multi-label classification scenarios since the number of classes per instance can vary significantly.

4 Method

As illustrated in Figure 1, our few-shot classification method consists of separate training parts for the model body and the model head. We fine-tune a sentence embedding model as the model body, while the model head consists of a simple logistic regression classifier trained on the data encoded by the model body.

4.1 Model Body

Given a base PLM, we fine-tune the model body of FusionSent in three steps: (1) use SetFit's contrastive learning approach to construct positive and negative training pairs from the few training examples to fine-tune a sentence embedding model from the base PLM (2) construct positive and negative training pairs from the few training examples and their corresponding label texts to fine-tune a different sentence embedding model using the same base PLM, and (3) merge both fine-tuned sentence embedding models to obtain the model body of FusionSent.

In the first step, we fine-tune a sentence embedding model from the base PLM using contrastive learning and the few training examples. Specifically, from the few training examples, instances of the same class are selected as positive pairs, which are assigned a score of 1, and instances from different classes are selected as negative pairs, which are assigned a score of 0. These training pairs are then used to fine-tune a sentence embedding model with the Cosine Similarity Loss:

$$L_{cos} = \|y - cos\_sim(u, v)\|_2, \tag{1}$$

where $u, v \in \mathbb{R}^D$ are the D-dimensional sentence embeddings of two sentences respectively, and $y \in \{0, 1\}$ is the pair label.

In the second step, we use a different contrastive training approach to fine-tune a separate sentence embedding model from the same base PLM, using the few training instances and their corresponding label texts. Specifically, positive pairs consist of training instances and the label texts of the class assigned to them. Negative pairs consist of training instances and label texts from different classes. Label texts can consist of simple label/class names, which are usually available in datasets, or of more extensive texts that semantically describe the meaning of a label/class. We assign the positive pairs a score of 1 and the negative pairs a score of 0 to fine-tune a sentence embedding model with the Contrastive Loss (Hadsell et al., 2006):

$$L_{cont} = \frac{1}{2}\big[y \cdot cos\_dist(u, v)^2 + (1 - y) \cdot max\{0, m - cos\_dist(u, v)\}^2\big], \tag{2}$$

where $u, v \in \mathbb{R}^D$ are the D-dimensional sentence embeddings of two sentences respectively, $m = 0.5$ is a margin, and $y \in \{0, 1\}$ is the pair label.

To obtain the contrastive training pairs for steps one and two, we use an *oversampling* strategy. In this approach, an equal number of positive and negative training pairs are sampled, with the minority pair type (positive) being oversampled to align with the majority pair type (negative).

In the third step, the parameters of the fine-tuned sentence embedding models obtained in steps one and two are merged using Spherical Linear Interpolation (SLERP) (Shoemake, 1985). Specifically, let θ_1 be the parameters obtained from the first fine-tuning step and θ_2 the parameters obtained from the second fine-tuning step, we merge parameters with SLERP:

$$slerp(\theta_1, \theta_2; t) = \frac{\sin(1-t)\Omega}{\sin\Omega}\theta_1 + \frac{\sin t\Omega}{\sin\Omega}\theta_2, \tag{3}$$

where $\theta_1 \cdot \theta_2 = \cos\Omega$ and $t = 0.5$ is an interpolation factor. Finally, the new parameters θ_3 obtained from SLERP merging are inserted into a sentence embedding model derived from the same architecture as the base PLM, resulting in the FusionSent model body.

4.2 Model Head & Inference

In the second part of FusionSent training, we first use the frozen model body to embed all available training instances. Then, we train a logistic regression model using the embedded training instances as input features. During inference, the model body embeds the inputs to provide features for the logistic regression head that subsequently classifies the unseen instances.

5 Experiments

5.1 Data

We construct a dataset of scientific documents derived from arXiv metadata (Clement et al., 2019). The arXiv metadata provides information about more than 2 million scholarly articles published in arXiv from various scientific fields. We use this metadata to create a dataset of 203,961 titles and abstracts categorized into 130 different classes. To this end, we first perform a stratified downsampling of the metadata to only 10% of all articles while retaining the original class distribution. Afterward, articles assigned to categories occurring less than 100 times in the downsampled dataset are removed. To obtain the final dataset, we then perform a stratified train/validation/test split of the processed dataset in an 80:10:10 ratio. The number of examples in each set are shown in Table 1.

Dataset split	Size
Train	163,168
Validation	20,396
Test	20,397

Table 1: Overview of the arXiv dataset.

Each article in the resulting arXiv dataset is categorized into one or more distinct categories. Figure 2 shows the distribution of papers across the 130 categories of the dataset.

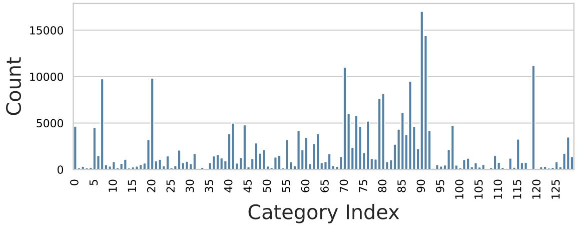


Figure 2: Number of papers in each category of the arXiv dataset.

In addition, we use the SciHTC dataset (Sadat and Caragea, 2022), which contains computer science papers categorized into one or more classes of the ACM Computing Classification System. We remove classes with less than 100 examples, resulting in 46,372 training samples and 5,838 test samples categorized into 62 different classes.

As a third dataset for scientific document classification, we use the NLP taxonomy dataset (Schopf et al., 2023), which contains papers from the ACL Anthology[4], the arXiv cs.CL category, and Scopus[5], categorized into one or more Natural Language Processing (NLP)-related classes. We perform a stratified 90:10 split between training and test examples, resulting in 161,414 training and 17,935 test instances categorized into 82 different classes.

5.2 Models

We experiment with two baselines and four different few-shot learning approaches for multi-label classification of scientific documents.

FineTune The first baseline consists of a standard encoder-only transformer that is fine-tuned for text classification.

SetFit The second baseline consists of the SetFit approach without any changes to the architecture or the training procedure.

Label Embedding (LE) As an initial few-shot learning approach, we experiment with only training one sentence embedding model that uses few-shot examples and their corresponding label texts in a contrastive learning approach. This approach consists of training a model body, as described in step 2 in Section 4.1, and training a logistic regression head on top of it.

SetFit→LE We also experiment with combining contrastive learning approaches to directly fine-tune a single sentence embedding model rather than separate models that are merged later. For this purpose, we perform training steps one and two as described in Section 4.1 sequentially on the same sentence embedding model. For classification, we then train a logistic regression head.

LE→SetFit This approach also only trains a single sentence embedding model. However, we first perform training step two followed by step one as described in Section 4.1 on the same model. We then train a logistic regression model head for the classification.

FusionSent Finally, we experiment with the FusionSent approach as described in Section 4.

5.3 Experimental Setup

Systematically evaluating few-shot performance is challenging due to the potential instability arising from fine-tuning on small datasets (Zhang et al.,

---

[4] https://aclanthology.org

[5] https://www.scopus.com

Dataset →	*arXiv*			*SciHTC*			*NLP Taxonomy*			*Average*
Method ↓	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
					$\lvert N\rvert = 2^*$					
FineTune	-	-	-	-	-	-	$13.1_{3.2}$	$78.3_{5.5}$	$7.2_{2.0}$	-
SetFit	$37.5_{1.6}$	$45.5_{1.7}$	$32.0_{2.4}$	$31.4_{0.8}$	$39.9_{0.1}$	$26.0_{1.2}$	$58.7_{2.0}$	$55.7_{4.2}$	$62.3_{0.9}$	$42.5_{1.5}$
Label Embedding	$43.2_{0.1}$	$45.3_{1.9}$	$40.4_{0.5}$	$34.1_{4.3}$	$46.1_{6.3}$	$27.1_{3.3}$	$65.8_{2.2}$	$66.0_{5.5}$	$66.0_{1.6}$	$47.7_{2.2}$
SetFit→LE	$41.5_{0.9}$	$45.3_{1.5}$	$38.3_{1.7}$	$32.3_{1.9}$	$41.6_{3.2}$	$26.3_{1.4}$	$64.6_{1.8}$	$62.7_{4.4}$	$66.8_{2.1}$	$46.1_{1.5}$
LE→SetFit	$39.2_{0.5}$	$41.5_{1.1}$	$37.1_{0.5}$	$34.0_{1.6}$	$38.9_{1.6}$	$30.2_{2.0}$	$61.3_{1.6}$	$55.0_{3.4}$	$69.3_{1.5}$	$44.8_{1.2}$
FusionSent	$\mathbf{44.4_{0.4}}$	$50.5_{1.2}$	$39.6_{0.1}$	$\mathbf{36.7_{1.9}}$	$48.0_{4.2}$	$29.7_{0.9}$	$\mathbf{66.2_{2.1}}$	$67.4_{4.6}$	$65.2_{0.8}$	$\mathbf{49.1_{1.5}}$
					$\lvert N\rvert = 4^*$					
FineTune	-	-	-	$10.3_{1.6}$	$59.3_{3.6}$	$5.6_{0.9}$	$43.5_{1.8}$	$85.2_{2.1}$	$29.2_{1.9}$	-
SetFit	$45.6_{1.0}$	$46.4_{1.6}$	$45.1_{0.9}$	$35.0_{0.3}$	$42.6_{1.1}$	$29.7_{0.3}$	$63.6_{1.5}$	$60.7_{2.3}$	$66.9_{1.8}$	$48.1_{0.9}$
Label Embedding	$47.0_{1.1}$	$47.7_{0.6}$	$46.3_{2.0}$	$32.7_{1.1}$	$41.9_{2.3}$	$26.9_{0.6}$	$71.9_{0.6}$	$70.8_{0.5}$	$73.0_{1.2}$	$50.5_{0.9}$
SetFit→LE	$46.0_{0.3}$	$47.4_{0.4}$	$44.7_{0.7}$	$30.7_{2.4}$	$36.9_{4.4}$	$29.5_{2.1}$	$70.0_{0.6}$	$68.1_{0.8}$	$72.1_{0.4}$	$48.9_{1.7}$
LE→SetFit	$45.8_{1.0}$	$45.0_{1.4}$	$46.7_{0.9}$	$32.0_{2.9}$	$40.6_{2.3}$	$27.8_{1.9}$	$66.1_{0.7}$	$62.1_{0.8}$	$70.7_{0.9}$	$48.0_{1.5}$
FusionSent	$\mathbf{48.3_{1.1}}$	$51.0_{1.0}$	$46.0_{1.5}$	$\mathbf{38.5_{2.3}}$	$45.4_{1.0}$	$33.5_{2.9}$	$\mathbf{72.6_{0.5}}$	$72.1_{0.3}$	$73.2_{0.7}$	$\mathbf{53.1_{1.3}}$
					$\lvert N\rvert = 8^*$					
FineTune	$18.7_{2.8}$	$72.5_{2.4}$	$10.8_{1.9}$	$26.6_{3.9}$	$55.0_{3.7}$	$17.5_{3.1}$	$67.1_{1.1}$	$88.0_{1.6}$	$54.2_{1.1}$	$37.5_{2.6}$
SetFit	$46.0_{0.6}$	$44.1_{0.5}$	$48.1_{1.1}$	$37.5_{3.7}$	$46.2_{3.2}$	$31.5_{3.8}$	$66.2_{0.2}$	$66.4_{0.4}$	$66.0_{0.4}$	$49.9_{1.5}$
Label Embedding	$47.8_{1.4}$	$46.3_{1.5}$	$49.5_{1.4}$	$36.4_{3.2}$	$41.3_{2.5}$	$32.5_{3.5}$	$77.2_{0.5}$	$74.0_{0.7}$	$80.8_{0.7}$	$53.8_{1.7}$
SetFit→LE	$45.3_{1.2}$	$45.5_{1.0}$	$45.1_{1.4}$	$30.5_{1.6}$	$31.4_{2.0}$	$29.7_{1.9}$	$72.7_{0.4}$	$70.5_{0.5}$	$75.1_{0.4}$	$49.5_{1.1}$
LE→SetFit	$44.1_{1.0}$	$42.1_{1.8}$	$46.3_{0.5}$	$32.9_{3.0}$	$41.8_{4.1}$	$27.1_{2.3}$	$66.7_{0.4}$	$65.8_{1.3}$	$67.6_{1.1}$	$47.9_{1.5}$
FusionSent	$\mathbf{49.0_{1.3}}$	$49.0_{1.8}$	$49.0_{0.9}$	$\mathbf{41.2_{4.6}}$	$43.4_{5.4}$	$39.2_{3.9}$	$\mathbf{78.3_{0.3}}$	$76.0_{0.2}$	$80.7_{0.5}$	$\mathbf{56.2_{2.1}}$
					$\lvert N\rvert = Full^{**}$					
FineTune	71.6	78.2	66.1	57.9	73.5	47.8	95.9	96.2	95.7	75.1

Table 2: FusionSent performance scores and standard deviations for few-shot classification of scientific documents compared to different approaches across three test datasets and four training set sizes $|N|$. Micro F_1, Precision (P), and Recall (R) scores are reported. *Number of training samples per class. **Entire available training data used. In some cases, insufficient training examples were provided for the FineTune model to learn, resulting in no class predictions during testing.

2021). In our multi-label scientific document classification experiments, we use three random training splits for each dataset and sample size to mitigate this issue. For each method, we report the average measure and the standard deviation across these splits. We use SciNCL (Ostendorff et al., 2022) as the base PLM for each model. While we train the FineTune model for 50 epochs in the few-shot setting, we use the same approach to train a model on the full training datasets for 3 epochs. In both cases, we use a batch size of 12. For sentence embedding model training according to step 1 in Section 4.1, we use a batch size of 4 for all models, and for training according to step 2 in Section 4.1, we use a batch size of 1. Both steps are trained for 1 epoch for all models. In addition, each model is trained with a learning rate of $2e^{-5}$.

For the arXiv dataset, we use the publicly available category descriptions as label texts. For the SciHTC and NLP taxonomy datasets, we generate short descriptive texts from the provided label names with GPT-4 (OpenAI et al., 2023) and use them as label texts. Table 5 shows examples of the used label names and label descriptions.

6 Results

Table 2 shows a comparison between FusionSent and the other few-shot approaches for $|N| \in \{2, 4, 8\}$ labeled training samples per class. We observe that FusionSent consistently outperforms F_1 scores of all approaches investigated for each dataset and training set size. Further, FusionSent significantly outperforms SetFit across all training set sizes by an average of 6.0 F_1 points. While the other approaches using label texts for sentence embedding training on a single model can perform better than SetFit, they fall short of the FusionSent approach. The Label Embedding (LE) approach shows consistent improvements over SetFit on average, while the SetFit→LE and LE→SetFit approaches only outperform SetFit in a few cases.

The results demonstrate that using label texts for sentence embedding training can help to separate instances of different classes in the embedding space, providing a crucial property for the classification head to perform well. However, combining the contrastive sentence embedding training approaches of SetFit and LE in a single model does not significantly increase performance. Using a two-step contrastive training approach does not enable a single sentence embedding model to effectively encode information from both training steps, as it may suffer from forgetting previously acquired knowledge (Biesialska et al., 2020). Conversely, the FusionSent results indicate that this limitation can be circumvented by training separate sentence embedding models with different contrastive learning approaches and subsequently merging their parameters. This approach ensures that the individually trained models encode different information, whereas merging allows their respective knowledge to complement each other, resulting in improved model performance.

7 Experiments with Few Classes

To determine the generalizability of FusionSent to few-shot settings with a low number of classes and different domains, we perform experiments on the SST-5 (Socher et al., 2013), CR (Hu and Liu, 2004), Emotion (Saravia et al., 2018), AGNews (Zhang et al., 2015), and EnronSpam (Metsis et al., 2006) datasets as processed for few-shot classification by Tunstall et al. (2022). These datasets each comprise two to six classes and cover the fields of sentiment classification, opinion detection from customer reviews, emotion detection from *Twitter* tweets, news article classification, and e-mail spam detection. For all datasets, we generate short descriptive texts from the provided label names with GPT-4 and use them as label texts. Table 5 shows examples of the used label names and label descriptions. We experiment with all ten available randomized training splits for each dataset and sample size using $|N| = 8$ and $|N| = 64$ few-shot examples. We report the average accuracy and standard deviation across the training splits for each method.

We use the paraphrase-mpnet-base-v2[6] model (Reimers and Gurevych, 2019) as base PLM for SetFit and FusionSent. Additionally, we use RoBERTa$_{LARGE}$ (Liu et al., 2019) for the FineTune baseline. The other training parameters remain the same as in Section 5.3.

Results Table 3 shows the results of SetFit and FusionSent training on the binary and multi-class datasets. On average, FusionSent outperforms SetFit for $|N| = 8$ by an average of 2.2 accuracy points. However, as the number of training samples increases to $|N| = 64$, the gap decreases to 0.6 accuracy points. In addition, the improvements are more substantial for multi-class classification, whereas they are only minimal in the binary case.

Dataset →	**SST-5**	**AGNews**	**Emotion**	**EnronSpam**	**CR**	**Average**
Method ↓	Multi-class Classification			Binary Classification		
			$\|N\| = 8^*$			
FineTune†	$33.5_{2.1}$	$81.7_{3.8}$	$28.7_{6.8}$	$85.0_{6.0}$	$58.8_{6.3}$	$57.5_{5.0}$
SetFit	$41.7_{2.0}$	$82.6_{3.6}$	$49.5_{3.8}$	$91.0_{3.2}$	$89.6_{1.2}$	$70.9_{2.8}$
FusionSent	$\mathbf{43.0_{3.2}}$	$\mathbf{84.4_{2.2}}$	$\mathbf{57.1_{2.5}}$	$\mathbf{91.4_{3.9}}$	$\mathbf{89.8_{1.0}}$	$\mathbf{73.1_{2.6}}$
			$\|N\| = 64^*$			
FineTune†	$45.9_{6.9}$	$88.4_{0.9}$	$65.0_{17.2}$	$95.9_{0.8}$	$88.9_{1.9}$	$76.8_{5.5}$
SetFit	$48.1_{4.4}$	$87.7_{0.8}$	$78.5_{2.0}$	$96.1_{0.5}$	$90.6_{0.7}$	$80.2_{1.7}$
FusionSent	$\mathbf{50.0_{2.8}}$	$\mathbf{88.3_{0.8}}$	$\mathbf{78.7_{1.6}}$	$\mathbf{96.5_{0.6}}$	$\mathbf{90.7_{0.6}}$	$\mathbf{80.8_{1.3}}$
			$\|N\| = Full^*$			
FineTune†	59.8	93.8	92.6	99.0	92.4	87.5

Table 3: FusionSent accuracy scores and standard deviations for few-shot classification with few classes compared to the baselines across five test datasets for three training set sizes. *Number of training samples per class. **Entire available training data used. †Results from Tunstall et al. (2022).

For binary classification, instances of different classes must be pushed apart from each other as far as possible to allow the logistic regression classifier to find a good decision boundary. For this relatively simple problem, the results indicate that SetFit can already effectively separate instances of different classes in embedding space. For multiple classes, however, it is more difficult to find positions in the embedding space that separate instances of different classes from each other. In these cases in particular, FusionSent shows its strengths and helps the classification head to find good decision boundaries between classes.

8 Robustness Against Label Text Variations

To evaluate the robustness of FusionSent against different label text variations, we conduct experiments on the previous datasets using simple label names instead of extensive label descriptions. In these experiments, we simply use the label names as provided by the respective datasets and compare the classification results with those obtained by using detailed label descriptions. Table 5 shows examples of the used label names and label descriptions. We use $|N| = 8$ few-shot examples and report the average F_1 performance over the respective training splits. Furthermore, we use the same training parameters as in Section 5.3 and Section 7.

Results Figure 3 shows the performance of FusionSent using simple label names, as provided by the respective datasets, compared to using exten-

---

[6] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

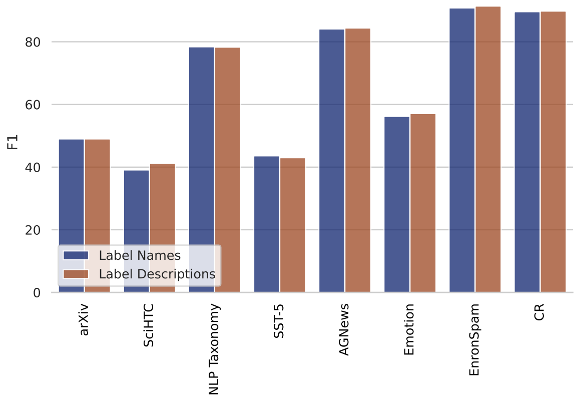


Figure 3: FusionSent micro F_1 scores for few-shot classification on 8 different datasets using either extensive label descriptions or simple label names. We report the average score over the random training splits of each dataset using $|N| = 8$ training examples per class.

sive textual label descriptions generated by GPT-4. We obtain similar performances across the different label text variants with a mean performance difference of 0.48 and a standard deviation of 0.39 F_1 points. Furthermore, there is no clear pattern as to whether the use of extensive label descriptions or simple label names leads to significantly improved performance. In comparison, performance variations of 10 accuracy points and more when using different prompts on the same model are characteristic for prompt-based few-shot classification approaches (Perez et al., 2021). Therefore, we conclude that FusionSent is relatively robust to label text variations and the use of simple label names is already sufficient to achieve good classification performance.

9 Computational Costs

To compare the relative computational costs of FusionSent and SetFit, we follow the approach of Liu et al. (2022) and use FLOPs-per-token estimates, which can be obtained from Kaplan et al. (2020). Specifically, encoder-only models with N parameters have approximately $2N$ FLOPs-per-token for inference and $6N$ FLOPs-per-token for training. The resulting cost for inference and training is then given by:

$$C_{inf} = 2N \cdot \ell_{seq}, \tag{4}$$

$$C_{train} = 6N \cdot \ell_{seq} \cdot \; n_{steps} \cdot n_{batch}, \tag{5}$$

where ℓ_{seq} is the input sequence length, n_{steps} is the number of training steps, and n_{batch} is the batch size. Since we are training two model bodies for FusionSent, we calculate the training costs for each model body separately and then add them up. For inference, we can use the formula as provided, since we only use one model body.

We estimate the costs using the scientific document classification datasets from Table 2 and SciNCL as base PLM with $N = 110M$ parameters. Based on the median number of tokens per instance in all datasets, we use $\ell_{seq} = 194$ to estimate the costs for training approaches that do not use label texts. Since we perform inference on these instances, we also use this value to estimate the inference cost for all approaches. Taking into account the shorter label texts, we use $\ell_{seq} = 130$ to estimate the costs for training approaches that utilize label texts. Additionally, we use fixed values of $n_{steps} = 1,000$, and $n_{batch} = 8$ for all training estimates.

Method	**Inf. FLOPs**	**Train FLOPs**	**Avg. $\mathbf{F_1}$**
SetFit	$4.3e10$	$1.0e15$	$49.9_{1.5}$
FusionSent	$4.3e10$	$1.7e15$	$56.2_{2.1}$

Table 4: Computational costs and average micro F_1 scores of FusionSent and SetFit using $|N| = 8$ training samples on the scientific document classification datasets listed in Table 2.

As shown in Table 4, the increase in F_1 performance is accompanied by increased training costs. This is the result of training two sentence embedding models instead of one. However, by merging the models, the inference efficiency remains the same as when using the base PLM. Although FusionSent incurs higher training costs, it can significantly improve prediction performance while maintaining SetFit's inference efficiency.

10 Conclusion

We introduce FusionSent, a new approach for efficient and prompt-free few-shot classification of scientific documents. FusionSent uses label texts and contrastive learning to improve classification performances over several other few-shot approaches. We show that FusionSent is particularly effective in scenarios with many classes while being computationally efficient during inference. Additionally, FusionSent is robust against label text variations. Finally, we introduce a new arXiv dataset for multi-label classification of scientific documents.

References

Rami Aly, Xingjian Shi, Kaixiang Lin, Aston Zhang, and Andrew Wilson. 2023. Automated few-shot classification with instruction-finetuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2414–2432, Singapore. Association for Computational Linguistics.

Luke Bates and Iryna Gurevych. 2024. Like a good nearest neighbor: Practical content moderation and text classification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 276–297, St. Julian's, Malta. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

D. R. Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.

Yuxiao Dong, Hao Ma, Zhihong Shen, and Kuansan Wang. 2017. A century of science: Globalization of scientific collaborations, citations, and innovations. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1437–1446, New York, NY, USA. Association for Computing Machinery.

Óscar E. Mendoza, Wojciech Kusa, Alaa El-Ebshihy, Ronin Wu, David Pride, Petr Knoth, Drahomira Herrmannova, Florina Piroi, Gabriella Pasi, and Allan Hanbury. 2022. Benchmark for research theme classification of scholarly documents. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 253–262, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Nikolaos Gialitsis, Sotiris Kotitsas, and Haris Papageorgiou. 2022. Scinobo: A hierarchical multi-label classifier of scientific publications. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 800–809, New York, NY, USA. Association for Computing Machinery.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Yongxin Huang, Kexin Wang, Sourav Dutta, Raj Patel, Goran Glavaš, and Iryna Gurevych. 2023. AdaSent: Efficient domain-adapted sentence embeddings for few-shot classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3420–3434, Singapore. Association for Computational Linguistics.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. volume 2 of *ICML deep learning workshop*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

N. Littlestone and M.K. Warmuth. 1994. The weighted majority algorithm. *Information and Computation*, 108(2):212–261.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Inf. Retr.*, 3(2):127–163.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA.

Kristine Monteith, James L. Carroll, Kevin Seppi, and Tony Martinez. 2011. Turning bayesian model averaging into bayesian model combination. In *The 2011 International Joint Conference on Neural Networks*, pages 2657–2663.

Ghulam Mustafa, Muhammad Usman, Lisu Yu, Muhammad Tanvir afzal, Muhammad Sulaiman, and Abdul Shahid. 2021. Multi-label classification of research articles using word2vec and identification of similarity threshold. Scientific Reports.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 11054–11070. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2022. Hierarchical multi-label classification of scientific documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8923–8937, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.

Angelo A. Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. 2019. The cso classifier: Ontology-driven detection of research topics in scholarly articles. In *Digital Libraries for Open Knowledge*, pages 296–311, Cham. Springer International Publishing.

António Paulo Santos and Fátima Rodrigues. 2009. Multi-label hierarchical text classification using the acm taxonomy. In *14th portuguese conference on artificial intelligence (epia)*, volume 5, pages 553–564. Springer Berlin.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Tim Schopf, Karim Arabi, and Florian Matthes. 2023. Exploring the landscape of natural language processing research. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1034–1045, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tim Schopf and Florian Matthes. 2024. NLP-KG: A system for exploratory search of scientific literature in natural language processing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 127–135, Bangkok, Thailand. Association for Computational Linguistics.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations*, pages 87–92, Melbourne, Australia. Association for Computational Linguistics.

Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, page 245–254, New York, NY, USA. Association for Computing Machinery.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Autumn Toney and James Dunham. 2022. Multi-label classification of scientific research documents across domains and languages. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 105–114, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Nan Xu, Fei Wang, Mingtao Dong, and Muhao Chen. 2023a. Dense retrieval as indirect supervision for large-space decision making. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15021–15033, Singapore. Association for Computational Linguistics.

Ran Xu, Yue Yu, Joyce Ho, and Carl Yang. 2023b. Weakly-supervised scientific document classification via retrieval-augmented multi-stage training. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2501–2505, New York, NY, USA. Association for Computing Machinery.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Asaf Yehudai and Elron Bendel. 2024. When llms are unfit use fastfit: Fast and effective text classification with many classes.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample {bert} fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Appendix

Dataset	Label Names	Label Descriptions
arXiv	General Relativity and Quantum Cosmology ...	General relativity and quantum cosmology focuses on gravitational physics, including experiments and observations related to the detection and interpretation of gravitational waves, experimental tests of gravitational theories, computational general relativity, relativistic astrophysics, solutions to Einstein's equations and their properties, alternative theories of gravity, classical and quantum cosmology, and quantum gravity. ...
SciHTC	Information retrieval ...	The "Information Retrieval" class within the 2012 ACM Computing Classification System encompasses the study and design of systems for indexing, searching, and retrieving information from large datasets. It includes the development of algorithms and techniques for processing and querying textual and multimedia data, as well as evaluating the effectiveness of retrieval systems. Key topics within this class involve search engine architectures, query representation, relevance feedback, and information extraction. The field also addresses challenges such as handling unstructured data, understanding user context, and ensuring privacy and security in the retrieval process. ...
NLP Taxonomy	Named Entity Recognition ...	Named Entity Recognition is the identification and classification of entities (e.g., names of people, organizations) in text. ...
SST-5	very positive ...	'very positive' is used for data samples that express strong or intense positive sentiments, enthusiasm, or approval. ...
AGNews	Sports ...	'Sports' represents data samples related to sports news, events, scores, and athlete performances. ...
Emotion	sadness ...	'sadness' is characterized by feelings of hopelessness, disappointment, melancholy, and vulnerability, often accompanied by a sense of isolation or being overwhelmed. ...
EnronSpam	spam ...	'spam' is an unsolicited and often irrelevant or inappropriate message sent over the internet, typically to a large number of users, for the purpose of advertising, phishing, spreading malware, or other malicious activities. ...
CR	negative ...	'negative' corresponds to criticisms, complaints, or expressions of dissatisfaction with products or services. ...

Table 5: Examples of label names and extensive label descriptions for different datasets.

A Comparison of Different Tokenization Methods for the Georgian Language

Beso Mikaberidze[†], **Teimuraz Saghinadze**[†], **Guram Mikaberidze**[‡],
Raphael Kalandadze[†], **Konstantine Pkhakadze**[¶], **Josef van Genabith**[*],
Simon Ostermann[*], **Lonneke van der Plas**[**], **Philipp Müller**[*]

[†]MICM Georgia, [‡]UWYO USA, [¶]GTU Georgia, [**]IDIAP Switzerland, [*]DFKI Germany

`beso.mikaberidze@gmail.com, philipp.mueller@dfki.de`

Abstract

While the impact of tokenization on language modeling is well-researched in richly resourced languages, fewer studies on this topic exist for challenging low-resource languages. In this work, we present the first systematic evaluation of tokenization methods for Georgian, a low-resource language with high morphological complexity. We compare standard subword tokenizers, such as WordPiece, Byte Pair Encoding, SentencePiece with Unigram, and a recently proposed token-free approach. We also investigate the multilingual BERT tokenizer (mBERT), which includes Georgian. In addition to these different classes of tokenization algorithms we also evaluate the impact of different vocabulary sizes, a key parameter for subword tokenizers. We evaluate the performance of all tokenizers on masked language modeling and on four downstream tasks: part-of-speech tagging, named entity recognition, toxicity detection, and sentiment analysis. We observe that larger vocabulary sizes for subword tokenizers generally lead to better performance across most tasks, with a notable exception in the toxicity detection task, where finer subword granularity is more effective. For the remaining tasks, pre-training tokenizers on Georgian text consistently yield better results compared to mBERT. Additionally, the token-free method is consistently outperformed by all other tokenizers. Taken together, our comprehensive evaluation of tokenizers will be highly valuable in making informed tokenization choices in future language model developments for Georgian.

1 Introduction

Tokenization is a fundamental process in most natural language processing (NLP) tasks that involves breaking down a text into smaller units called *tokens*. It is one of the first processes conducted in most approaches and is particularly crucial for low-resource languages. Tokenization gains further importance in morphologically complex languages where multiple types of prefixes and suffixes simultaneously modify the meaning of a word, making it vital to split each word into meaningful pieces. That is why different tokenization methods have been investigated in languages such as Turkish (Toraman et al., 2023), Arabic (Alyafeai et al., 2023), or Korean (Park et al., 2020). Studies on these languages have shown that appropriate tokenization can significantly enhance model performance, with subword-level tokenization often providing a good balance between capturing linguistic nuances and managing sequence lengths.

In contrast, no comprehensive study of tokenization has been conducted for any of the languages from the Kartvelian family to which Georgian belongs. The Kartvelian family has no known relation to any other language group. It consists of four languages, all spoken in Georgia, with its first split dating back to the 20-22th century BC (Gavashelishvili et al., 2023). Georgian, the official language of Georgia, serves as a common language for all Kartvelian speakers. The language is phonetic and is written in its unique alphabet, one of the world's approximately 15 base alphabetical systems. Georgian, a low-resource language with complex morphology, has seen limited progress in NLP research, which remains in its early stages. Existing studies have primarily focused on data curation (Beridze et al., 2017; Stefanovitch et al., 2022a) and syntactic and morphological analysis (Kapanadze, 2019; Kardava et al., 2017; Lobzhanidze, 2022) rather than tokenization. Conducting a comprehensive evaluation of tokenizers for Georgian provides a solid foundation for future research on building effective Georgian language models, addressing its unique linguistic challenges, and improving NLP applications.

In our work, we address this need by, for the first time, systematically evaluating different tokenizers in Georgian for language modeling and on a set of four downstream tasks. In particular, we evaluate four tokenization techniques: WordPiece (Song et al., 2021), Byte Pair Encoding (Sennrich et al., 2016), SentencePiece with Unigram (Kudo and Richardson, 2018, Kudo, 2018), and a token-free method (Xue et al., 2022). With these tokenizers, we train a scaled-down BERT (Devlin et al., 2018) architecture on a substantial Georgian language corpus and fine-tune it on four downstream applications: sentiment analysis, toxicity detection, named entity recognition, and part-of-speech tagging. In addition, we investigate various vocabulary sizes by training different-sized tokenizer models, identifying optimal strategies tailored to Georgian's morphological characteristics. Our results indicate that (1) subword tokenization approaches trained on Georgian pretraining corpora are superior to the token-free approach as well as multilingual BERT's WordPiece tokenizer, and (2) that larger vocabulary sizes tend to improve performance. The main exception is the toxicity detection task, where tokenizers with finer granularity perform better. These include multilingual BERT's WordPiece with its smaller vocabulary as well as the smaller vocabulary versions of the subword tokenizers. With our approaches, we set a new state of the art on the recently introduced toxicity detection dataset by Lashkarashvili and Tsintsadze (2022).

The source code developed in this study is available online[1].

2 Related Work

2.1 Tokenizers in Language Modelling

We distinguish three major categories of tokenizers: word-level, subword-level, and token-free (character/byte-level tokenizers).

Word-level tokenizers take all distinct words in the corpus as tokens, which results in large vocabularies that are, however, still rarely exhaustive. While not requiring specific training, such tokenizers often suffer from numerous out-of-vocabulary cases (Luong et al., 2015).

Subword-level tokenization is the most common tokenization technique for modern language models. Such tokenizers are trained and selectively combine characters, subwords, and words. Words that are rarely used are usually split into smaller units, resulting in smaller vocabulary sizes at better coverage and fewer out-of-vocabulary cases.

GPT 2, 3 and RoBERTa (Radford et al., 2019, Brown et al., 2020, Liu et al., 2019) utilize a Byte Pair Encoding (BPE) tokenization method (Sennrich et al., 2016). BERT and ELECTRA (Devlin et al., 2018, Clark et al., 2020) use a variant of the BPE, the WordPiece tokenization method (Song et al., 2021). XLM-RoBERTa, XLNet, and T5 (Conneau et al., 2020, Yang et al., 2019, Raffel et al., 2020), all rely on SentencePiece (Kudo and Richardson, 2018) with the Unigram algorithm Kudo (2018).

Token-free approaches treat all distinct characters or bytes in the corpus as tokens, resulting in a small vocabulary and no out-of-vocabulary cases, but also significantly longer input sequences and less meaningful individual tokens. Byte-level tokenizers have been shown to be competitive with their subword-level counterparts but usually need more training time (Xue et al., 2022).

2.2 Tokenization for Morphologically Rich and Low-Resource Languages

Toraman et al. (2023) show that for languages with rich morphology, the choice of tokenizer can significantly affect model performance. Word-level tokenization often struggles due to the large number of possible word forms, whereas subword-level tokenizers and token-free approaches can provide more flexibility and robustness by capturing meaningful subunits and handling out-of-vocabulary words effectively. Similarly, Park et al. (2020) discuss the importance of appropriate tokenization for Korean, a language with agglutinative morphology. They highlight how different tokenization strategies, such as character-level and subword-level, affect the performance of NLP models on diverse tasks and show that subword-level tokenization strikes a balance between capturing linguistic nuances and maintaining manageable sequence lengths. Alyafeai et al. (2023) examine how different tokenization methods perform on Arabic text classification tasks. Given the rich morphology and script variations, they show that tokenizers that can effectively handle these complexities are required. Subword-level tokenization, in particular, has been shown to provide better performance by capturing root and pattern morphemes.

2.3 Georgian Natural Language Processing

Georgian, a highly inflectional and agglutinative language with complex morphology, poses unique challenges for tokenization. Kartvelian, primarily

---

[1] `https://git.opendfki.de/philipp.mueller/icnlsp24`

spoken in Georgia, has no known relation to any other language groups, making it one of the world's primary language families.

Research on Georgian NLP is still in its early stages and to the best of our knowledge, no existing study focuses on tokenization methods. The majority of research has concentrated on data curation (Beridze and Nadaraia, 2009; Doborjginidze and Lobzhanidze, 2016; Fkhakadze et al., 2017; Beridze et al., 2017; Stefanovitch et al., 2022a) and automated syntactic and morphological analyzers (Kapanadze et al., 2019; Kapanadze, 2019; Kardava et al., 2017; Lobzhanidze, 2022). Some studies have trained models for downstream applications (Khachidze et al., 2016; Lashkarashvili and Tsintsadze, 2022; Stefanovitch et al., 2022a), using standard tokenization techniques without exploring the impact of tokenizers on the model's performance. Several papers (Pires et al., 2019, Conneau et al., 2020) with pre-trained multilingual language models provide subword-level tokenizers containing Georgian tokens. However, the current state of research indicates a gap in understanding how different tokenizers would perform for Georgian.

While subword-level tokenization has proven effective for large language models, even for morphologically rich languages like Turkish, Korean, and Arabic, the question of which subword-level algorithm would be most effective for Georgian remains open. The competitiveness of the token-free approach is also uncertain.

3 Data

In the following Section we present pre-training datasets and downstream task datasets used in our study.

3.1 Pre-training Datasets

In this work, we ensured a comprehensive coverage of the various styles and contexts of Georgian. We used three primary corpora to pre-train our tokenization models: Wikipedia [2], Leipzig (2016) [3], and CorpusGE (Fkhakadze et al., 2017). Wikipedia and Leipzig provide extensive text data across various domains, ensuring diverse language coverage. CorpusGE, a high-quality text corpus, was collected over four years from well-known Georgian media pages. Following previous work on Maltese (Micallef et al., 2022), we employed one million words from these corpora to pre-train our models.

	NER	POS	TOXD	SA
Epochs	10	30	10	10
Max. length	512	512	512	512
Batch size	384	24	192	192
Learning rate	2e-5	2e-5	2e-5	2e-5
Train Size	90,000	2,000	8,000	2,500
Val. Size	90,000	250	1,000	850
Test Size	92,000	250	1,000	850

Table 1: Training details for the four different tasks: Named-Entity Recognition (NER), Part-of-Speech Tagging (POS), Toxicity Detection (TOXD), and Sentiment Analysis (SA). In train and test sizes, we provide labeled word counts for token classification and labeled sentence counts for text classification tasks. **Exception:** Maximum length for token-free ByT5 is equal to 2048.

3.2 Downstream Tasks

To assess the performance of different tokenization methods, we focused on four language understanding tasks: two for text classification and two for token classification. We present an overview over downstream task dataset sizes in Table 1.

Named Entity Recognition (NER) Named Entity Recognition is a token classification task that identifies person, organization, or location names in the text. We utilized the Wikiann (pan-x) multilingual benchmark (Pan et al., 2017), a comprehensive dataset that includes Georgian. This benchmark, which consists of approximately 30,000 Georgian sentences and roughly 90,000 labeled words per train, validation, and test splits, provides a thorough dataset for NER.

Part-of-Speech Tagging (POS) Part-of-Speech Tagging is a token classification task that detects parts of speech with respect to each word in a sentence, such as nouns, verbs, adjectives, etc. We employed the Universal Dependencies dataset (Nivre et al., 2020) for Georgian, which contains approximately 2,500 words and 152 sentences. The dataset was split, with 10% used for validation and 10% for testing. We also provide the percentage break-

---

[2] `https://dumps.wikimedia.org/backup-index.html`

[3] `https://wortschatz.uni-leipzig.de/en/download/Georgian`

down of fourteen imbalanced class distributions: Noun (29%), Punc (14%), Adj (13%), Verb (9%), Pron (8%), Post (7%), Conj (6%), Adv (6%), Aux (3%), Part (2%), Prop (2%), Num (1%), VerbalAdj (0.3%), VerbalNoun (0.1%).

Toxicity Detection (TOXD) Toxicity detection is a text classification task identifying harmful or toxic comments in online discussions. For this task, we used a dataset provided by Lashkarashvili and Tsintsadze (2022). This data was gathered from Georgian online discussion forums and manually annotated for toxicity. The dataset comprises 10,000 sentences, divided into 46% toxic and 54% non-toxic samples. We split the train, validation, and test datasets as follows: 80%, 10%, and 10%.

Sentiment Analysis (SA) Sentiment Analysis is a text classification task that determines the emotional tone of the text. For this task, we used the first publicly released annotated sentiment dataset for Georgian (Stefanovitch et al., 2022b), referred to as Georgian Sentiment Snippets (GSS). This dataset contains approximately 4K text snippets, each manually annotated by multiple annotators using a four-tier scale: positive (33.5%), neutral (41.0%), negative (18.1%), and mixed (7.2%). The dataset is split into training, validation, and test sets with the following proportions: 60%, 20%, and 20%. This annotated dataset provides a robust resource for training and evaluating sentiment analysis models.

4 Approach

We first discuss the different tokenizers we compare in our study and subsequently present the training procedure we utilized.

4.1 Tokenizers

In this study, we compare various tokenization methods for Georgian. We focus on subword-level tokenizers, including WordPiece, Byte Pair Encoding (BPE), and SentencePiece with Unigram. Additionally, we explore the byte-level token-free approach ByT5, assessing its performance relative to traditional subword-level tokenizers.

Byte Pair Encoding (BPE) This approach was initially introduced for data compression (Sennrich et al., 2016). BPE minimizes the total number of symbols (characters or bytes) needed to represent the original text. First, the data is split into individual symbols. Then, the most frequent adjacent pairs of symbols are consecutively merged until the desired vocabulary size is reached. In this study, we employ BPE, which considers every distinct byte as its initial vocabulary.

WordPiece (WP) WordPiece (Song et al., 2021) is a variant of the BPE method. The primary difference lies in the merge rule, which is based on likelihood rather than solely on frequency. Specifically, the algorithm prioritizes token pairs that have a higher joint probability of how frequently the tokens appear together compared to how frequently they appear separately. This method aims to retain more meaningful linguistic units, potentially providing a more nuanced tokenization. However, training requires more computational resources due to the complexity of calculating these probabilities.

SentencePiece with Unigram (SP-U) The SentencePiece (Kudo and Richardson, 2018) is a tool that implements both the BPE and Unigram (Kudo, 2018) algorithms. This approach enables the tokenization of raw text strings without the need for preprocessing, such as whitespace splitting, making it particularly effective for languages without clear word boundaries.

The Unigram algorithm, employed within the SentencePiece framework, operates in two stages. First, it populates its vocabulary with a large number of tokens similar to BPE, but for searching the most frequent substrings, it uses the enhanced suffix array algorithm. Second, it decreases the vocabulary to the desired size. The Unigram model iteratively prunes the least likely tokens based on their probability contribution to the corpus, leveraging the expectation maximization (EM) algorithm.

Token-Free Byt5 This approach treats all distinct bytes in the corpus as tokens. Xue et al. (2022) used this approach and have increased the number of transformer parameters at the expense of a large number of discarded vocabulary parameters. They have been shown to be competitive with their subword-level counterparts.

Multilingual BERT To provide a comprehensive evaluation, we compare our pre-trained tokenizers with out-of-the-box multilingual BERT's (Pires et al., 2019) WordPiece tokenizer, containing 700 Georgian tokens. This comparison allows us to assess the effectiveness of our tokenizers against the established multilingual model.

4.2 Training Procedure

Tokenizer Training As the vocabulary size is a critical factor for the subword tokenizers, we ensure its optimization for each method. Each subword-level tokenizer was trained to generate vocabularies of four different sizes (8k, 16k, 32k, and 64k), ensuring optimal performance for the BERT model. All the tokenizers were adjusted to accommodate BERT's special tokens and post-processing requirements.

BERT Integration For the integration with BERT, we followed related studies (Toraman et al., 2023; Xue et al., 2022) and utilized a scaled-down architecture. These studies indicated that differences between tokenizers are more pronounced with smaller language models. Smaller models also have the advantage of faster training, allowing us to run more evaluations than would be possible with larger-scale models. For our scaled-down BERT model, we used the following configuration, consistent across all our experiments: Hidden size: 512; Number of hidden layers: 8; Number of attention heads: 8; Intermediate-size: 3072; Max position embeddings: 512 for subword-level tokenizers, 2048 for token-free approach.

Pre-training Setup The pre-training corpus, as detailed in Sec. 3, comprises 1 million tokens from high-quality Georgian text sources. Pre-training was conducted by training multiple BERT models sufficiently long to achieve stable training and evaluation loss plots. BERT models were pre-trained using only the Masked Language Modeling (MLM) task, with the following aspects deviating from the original BERT configuration. We made use of dynamic masking adopted from RoBERTa, set the training epochs to 30, the batch size to 264, and employed mixed precision training.

Finetuning and Evaluation The pre-trained BERT models were finetuned on four downstream language understanding tasks: Named Entity Recognition (NER) and Part-of-Speech (POS) tagging for token classification, and Sentiment Analysis and Toxicity Detection for text classification. Details on these tasks and their corresponding datasets are provided in Section 3. Each language model was finetuned 26 times, and evaluation results were averaged across these runs to ensure stability and robustness. Performance was evaluated using four metrics: accuracy, f1 score, precision, and recall. These metrics provide a comprehensive view of the models' effectiveness across the various tokenization methods.

		ByT5	**mWP**	**BPE**	**WP**	**SP-U**
MLM	**acc**	0.423	0.564	0.613	0.616	**0.617**
NER	**acc**	0.800	0.902	0.925	0.927	**0.930**
	f1	0.552	0.758	**0.797**	0.794	0.787
	pre	0.565	0.744	0.781	**0.783**	0.774
	rec	0.539	0.774	**0.813**	0.806	0.800
TOXD	**acc**	0.879	**0.955**	0.917	0.933	0.941
	f1	0.866	**0.952**	0.911	0.928	0.937
	pre	0.890	**0.948**	0.912	0.923	0.929
	rec	0.843	**0.957**	0.910	0.933	0.945
POS	**acc**	0.699	0.889	0.900	0.905	**0.915**
	f1	0.045	0.709	0.817	**0.824**	0.820
	pre	0.028	0.670	0.788	**0.795**	0.790
	rec	0.121	0.754	0.849	**0.856**	0.852
SA	**acc**	0.493	0.588	0.672	0.668	**0.675**
	f1	0.472	0.558	0.642	0.641	**0.647**
	pre	0.470	0.535	0.637	0.649	**0.663**
	rec	0.493	0.588	0.672	0.668	**0.675**

Table 2: Performance of different tokenizers across various NLP tasks in terms of accuracy, f1 score, precision, and recall. Tokenizers: ByT5, multilingual BERT's WordPiece (mBERT), Byte Pair Encoding (BPE), WordPiece (WP), SentencePiece with Unigram (SP-U). Tasks: Masked Language Modeling (MLM), Named-Entity Recognition (NER), Part-of-Speech Tagging (POS), Toxicity Detection (TOXD), and Sentiment Analysis (SA).

5 Results

5.1 Comparing Tokenizers

We present the results of different tokenizers on language modeling and our four downstream tasks in Table 2. All subword tokenizers in this table were trained with a vocabulary size of 64k. For masked language modelling, SentencePiece with Unigram (SP-U) achieves the highest accuracy of 0.617, closely followed by WordPiece (0.616 acc), and BPE (0.613). Both multilingual BERT's WordPiece tokenizer and the token-free ByT5 are

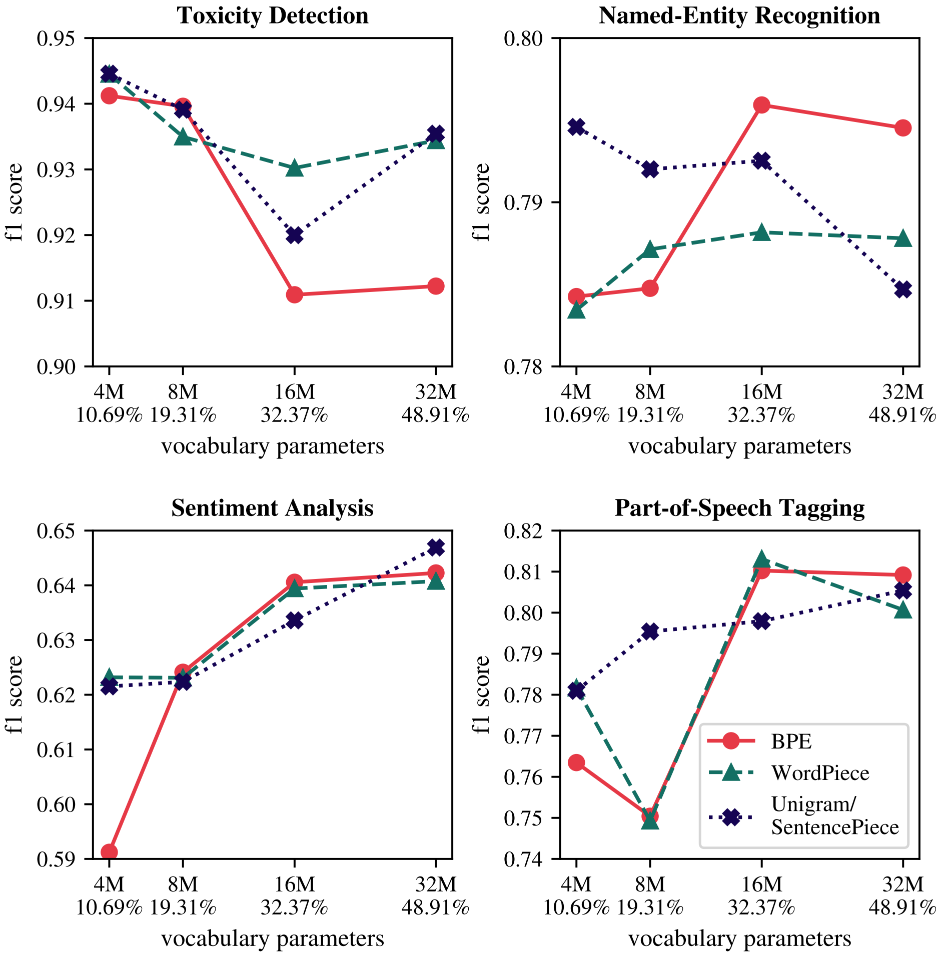


Figure 1: Impact of vocabulary size on the performance of four downstream tasks: Toxicity Detection, Sentiment Analysis, Named-Entity Recognition, and Part-of-Speech Tagging. The x-axis shows the vocabulary size in absolute numbers as well as in proportion to the overall network parameters.

worse by a large margin (0.56 and 0.423 acc, respectively). This general pattern is also present in three out of four downstream tasks. For named entity recognition, part-of-speech-tagging, and sentiment analysis, the subword tokenizers consistently achieve better performance than ByT5 and multilingual BERT's WordPiece tokenizer. The differences between subword tokenizers on downstream tasks are small. When measured in terms of f1, BPE achieves the best performance in named entity recognition (0.797 f1). For part-of-speech tagging, WordPiece achieves the best f1 score of 0.824, and for sentiment analysis, SentencePiece with Unigram is leading with 0.647 f1.

We observed surprisingly bad POS results for the ByT5 tokenizer in terms of f1 (0.045), precision (0.028), and recall (0.121). We conjecture this is because f1, precision, and recall are directly related to the number of correctly predicted positive instances. Because the tokenizer breaks the text into tokens that are too granular or not meaningful for the POS tagging task, combined with a small number of training examples, there is a high number of false positives and false negatives, thereby lowering the aforementioned metrics. Also, our POS tagging benchmark is highly imbalanced and involves a few frequent tags, like nouns and verbs, and many infrequent ones, like rare numerals. Thus, a high accuracy is misleading to some extent as the model performs well on frequent tags while failing on the rare ones.

5.2 Comparing Vocabulary Sizes

We present the results of our vocabulary size experiments in Figure 1. There is a tendency that larger vocabulary sizes lead to better performance. This is clearly the case for both sentiment analysis and part-of-speech tagging. For named entity recognition, the effect of vocabulary sizes is negligible - f1

	Accuracy	**AUC**
SOTA CNN	0.888 ± 0.007	0.942 ± 0.005
Ours (WP 8k)	0.944 ± 0.007	0.944 ± 0.007
Ours (mWP)	$\mathbf{0.959} \pm \mathbf{0.009}$	$\mathbf{0.959} \pm \mathbf{0.009}$

Table 3: Our Toxicity Detection approaches compared with the SOTA by Lashkarashvili and Tsintsadze (2022). We report accuracy and area under curve (AUC), along with standard deviations across CV folds.

scores only vary between 0.78 and 0.80. However, for toxicity detection, the positive connection between vocabulary size and performance is clearly reversed. Here, tokenizers with a higher average split of the words were more effective. This indicates that a finer granularity in tokenization can be beneficial for tasks requiring a nuanced understanding of potentially offensive language. This statement is in line with the previously observed fact that the multilingual BERT (mBERT) tokenizer performs best for toxicity detection. The mBERT tokenizer contains only 700 Georgian tokens, the smallest vocabulary size among the subword-level tokenizers we investigated.

5.3 Comparison with SoTA Approaches

Our scaled-down BERT models (42M parameters) demonstrate strong performance on the Toxicity Detection dataset introduced by Lashkarashvili and Tsintsadze (2022). We employed two tokenization methods for pretraining and fine-tuning: an 8K vocabulary WordPiece and the multilingual BERT WordPiece. For comparability, we followed Lashkarashvili and Tsintsadze (2022) by using stratified 5-fold cross-validation, along with accuracy (ACC) and area under the curve (AUC) as evaluation metrics. Results, presented in Table 3, show that while Lashkarashvili and Tsintsadze (2022) reported an ACC of 0.888 and an AUC of 0.942 for their best-performing CNN model, our approach achieved an ACC of 0.9435 and an AUC of 0.9442 using the 8K WordPiece, and an accuracy of 0.9586 and an AUC of 0.9591 with the multilingual WordPiece, establishing a new state of the art.

6 Discussion

For most tasks, we observed that pre-training tokenizers on a small amount of Georgian text yield better performance than relying on the mBERT tokenizer. This suggests that language-specific pretraining is crucial for achieving optimal results in Georgian NLP tasks. The superior performance of these tokenizers compared to the multilingual WordPiece tokenizer from mBERT (except in toxicity detection) raises questions about the limitations of the latter. Our findings indicate that this multilingual tokenizer may not adequately capture the nuances of highly divergent languages such as Georgian.

Furthermore, our results indicate that ByT5 is not competitive with the other tested methods. We suspect two possible reasons for this. First, each Georgian letter contains 3 bytes, so the LM training input sequences for Georgian are three times longer than for English. Second, in the original ByT5 paper, the authors Xue et al. (2022) increased the number of transformer parameters at the expense of a large number of discarded vocabulary parameters. They increased input sequence length, embedding size, and intermediate layer size. We only increased the input sequence length due to the limited vocabulary parameters, which might be another reason for the suboptimal performance observed.

We found a general trend of improved performance with larger vocabulary sizes for subword tokenizers. This suggests that capturing a wide range of morphological variations is crucial for effective language modeling in Georgian. However, our findings on toxicity detection versus the other downstream tasks also underscore the importance of tailoring tokenization strategies to the specific requirements of each task and dataset.

In our case, the toxicity detection benchmark involves words that are not present in the tokenizer's vocabulary, specifically those that serve as key indicators of toxic content. When a tokenizer encounters these unknown words, it splits them into smaller subword units. This behavior is observed even in tokenizers with large vocabularies. However, LMs using tokenizers with smaller vocabulary sizes are inherently more robust at handling and representing short tokens because their pretraining data mostly contains short tokens. In contrast, LMs using tokenizers with larger vocabularies tend to rely on longer tokens, which can lead to a loss of information when the input is split into less meaningful or less frequent short tokens. We conjecture that this is the reason for why tokenizers with smaller vocabulary sizes perform better in the case of toxicity detection.

The results highlight that a one-size-fits-all approach to tokenization is inadequate, and careful consideration must be given to the nature of the task and especially to the linguistic features of a language.

7 Conclusion

In this study, we explored the impact of various tokenization methods on Georgian language modeling, including subword-level tokenizers, such as BPE, WordPiece, and SentencePiece with Unigram, a pre-trained multilingual BERT tokenizer, and a recently proposed token-free approach ByT5. Each method is evaluated by the performance of a scaled-down BERT architecture on four independent downstream tasks. Our findings suggest that larger vocabulary sizes generally enhance performance across most NLP tasks. However, on the toxicity detection task, tokenizers with finer granularity, like the multilingual mBERT with its smaller vocabulary, performed better. In all the other tasks, language-specific pre-training of tokenizers outperformed mBERT. Interestingly, the token-free approach did not perform competitively, highlighting potential limitations of its applicability to Georgian, our model's architecture, or both.

In the future, we aim to explore the impact of different tokenization strategies on more advanced model architectures, as well as extend this analysis to other Kartvelian languages, which could further our understanding of effective NLP strategies for Georgian and similar languages.

Georgian presents a challenging landscape for NLP due to its complex morphology, limited training data, and sparse research focus. By conducting the first rigorous comparative study of tokenization methods for Georgian, this work lays a foundational reference for future research and development. Given that tokenization is the first step in NLP model training, our study provides valuable insights that can guide researchers and practitioners in building models tailored to the needs of Georgian and specific NLP tasks.

Limitations

While this study provides valuable insights into tokenization methods for Georgian, several limitations should be acknowledged.

Architectural Diversity Our research is limited to a scaled-down BERT. Exploring and experimenting with other LM architectures could potentially yield different results and even trends, and thus, it is essential to consider alternative architectures in future studies.

Language Scope The experiments and analyses conducted in this study are restricted to Georgian. Testing the generalizability of our findings to other languages would provide a broader validation of the tokenization methods. This is particularly important for ensuring the robustness and applicability of our approaches in multilingual contexts.

Downstream Tasks Our study evaluates the tokenization methods on a limited number of downstream tasks. Expanding the range of downstream tasks in future research will help to understand the effectiveness and limitations of the tokenization methods in diverse applications, potentially uncovering further task-specific strengths and weaknesses.

Comparison with Multilingual Models While we compared our tokenizers to the multilingual BERT model, we did not include XLM-RoBERTa and mT5 (Conneau et al., 2020; Xue et al., 2021) tokenizers in our evaluation. Future work should incorporate this and other recent multilingual models to provide a more complete comparison.

Acknowledgements

This work was partially supported by the European Union under Horizon Europe project *"GAIN"* (GA #101078950) and *DisAI - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies* (GA #101079164), as well as the BMBF-funded project TRAILS (01IW24005).

References

Zaid Alyafeai, Maged S. Al-shaibani, Mustafa Ghaleb, and Irfan Ahmad. 2023. Evaluating various tokenizers for arabic text classification. *Neural Processing Letters*, 55(3):2911–2933.

Marina Beridze, David Nadaraia, and Lia Bakuradze. 2017. Georgian dialect corpus: Linguistic and encyclopedic information in online dictionaries. *Journal of Linguistics/Jazykovedný casopis*, 68(2):109–121.

Marine Beridze and David Nadaraia. 2009. The corpus of georgian dialects. *NLP, Corpus Linguistics, Corpus Based Grammar Research*, page 25.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nino Doborjginidze and Irina Lobzhanidze. 2016. Corpus of the georgian language. In *Proceedings of the XVII EURALEX International Congress*, pages 328–335.

Konstantine Fkhakadze, Merab Chikvinidze, Giorgi Chichua, Davit Kurtskhaliya, and Inga Beridze. 2017. Georgian internet and web corpus.

Alexander Gavashelishvili, Merab Chukhua, Kakhi Sakhltkhutsishvili, Dilek Koptekin, and Mehmet Somel. 2023. The time and place of origin of south caucasian languages: insights into past human societies, ecosystems and human population genetics. *Scientific Reports*, 13(1):21133.

Oleg Kapanadze. 2019. Parsing the less-configurational georgian language with a context-free grammar. *Proceedings of the Language Technologies for All (LT4All), European Language Resources Association (ELRA), Paris, UNESCO Headquarters*, pages 342–345.

Oleg Kapanadze, Gideon Kotzé, and Thomas Hanneforth. 2019. Building resources for georgian treebanking-based nlp. In *International Tbilisi Symposium on Logic, Language, and Computation*, pages 60–78. Springer.

Irakli Kardava, Nana Gulua, Jemal Antidze, and Beka Toklikishvili. 2017. Morphological synthesis and analysis of georgian words.

Manana Khachidze, Magda Tsintsadze, and Maia Archuadze. 2016. Natural language processing based instrument for classification of free text medical records. *BioMed research international*, 2016(1):8313454.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Nineli Lashkarashvili and Magda Tsintsadze. 2022. Toxicity detection in online georgian discussions. *International Journal of Information Management Data Insights*, 2(1):100062.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Irina Lobzhanidze. 2022. Computational modeling. In *Finite-State Computational Morphology: An Analyzer and Generator for Georgian*, pages 117–166. Springer.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Kurt Micallef, Albert Gatt, Marc Tanti, Lonneke van der Plas, and Claudia Borg. 2022. Pre-training data quality and quantity for a low-resource language: New corpus and BERT models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 90–101.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An empirical study of tokenization strategies for various Korean NLP tasks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 133–142, Suzhou, China. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicolas Stefanovitch, Jakub Piskorski, and Sopho Kharazi. 2022a. Resources and experiments on sentiment classification for georgian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1613–1621.

Nicolas Stefanovitch, Jakub Piskorski, and Sopho Kharazi. 2022b. Resources and experiments on sentiment classification for Georgian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1613–1621, Marseille, France. European Language Resources Association.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinüç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.

GemST: Continual Learning for End-to-End Speech-to-Text Translation

Pranav Karande
IIT Indore
pranav.3943@gmail.com

Balaram Sarkar
IIT Indore
balarakar@gmail.com

Chandresh Kumar Maurya
IIT Indore
ckm.jnu@gmail.com

Abstract

Effective cross-lingual communication remains a significant challenge in today's rapidly globalizing world. Developing Speech-to-Text Translation (S2T) systems using artificial intelligence presents various difficulties, such as the unavailability of all language pairs for simultaneous model training. Additionally, when a model is trained on a new language, it often loses its ability to remember previously learned tasks, a phenomenon known as catastrophic forgetting. This paper explores the application of Gradient Episodic Memory (GEM) to address these challenges. Our study investigates the effectiveness of GEM in enhancing S2T model performance across sequentially introduced language pairs. Experimental results demonstrate that GEM can significantly reduce forgetting by **24.8%** and boost translation accuracy by **44.5%** as compared to baseline, offering a promising approach for scalable and efficient multilingual-continual S2T systems.

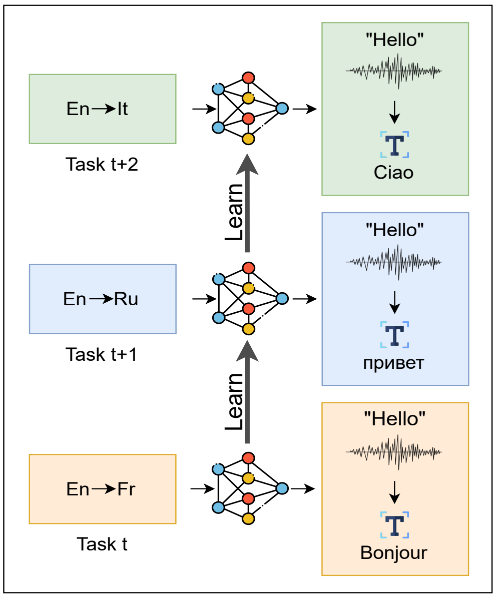


Figure 1: Task diagram of proposed work where new task (S2T for English to Russian, t+1) is trained using the model from previous task (S2T for English to French, t). And similarly for English to Italian (task t+2 trained from model of task t+1).

1 Introduction

Speech-to-text (S2T) translation is a technology that bridges language barriers by converting spoken language into written text in a different language. This capability is increasingly vital in our globalized world, where effective and seamless communication across diverse linguistic communities is essential. Traditional S2T translation systems like (Bansal et al., 2017; Le et al., 2021; Sarkar et al., 2023) typically require large, diverse datasets for training and are often retrained from scratch whenever new language pairs are introduced. This process is not only computationally expensive and time-consuming but also environmentally unsustainable due to the high energy consumption involved.

Continual learning, also known as lifelong learning, offers a promising solution to these challenges. In the realm of S2T translation, continual learning allows models to adapt incrementally to new languages and dialects without forgetting previously learned ones (Bremner et al., 2013; Rusu et al., 2022). Traditional methods suffer from Catastrophic Forgetting (McCloskey and Cohen, 1989), where a model's performance on previously learned tasks degrades as it learns new ones. Continual learning mitigates this issue by enabling S2T models to retain past knowledge while incorporating new information, thus maintaining high performance across all tasks. This approach not only improves the efficiency and scalability of multilingual S2T systems but also ensures they remain effective as new language data becomes available.

To this end our contributions include studying continual learning for end-to-end (E2E) S2T to mitigate the catastrophic forgetting. To the best of our knowledge, no prior research has been conducted on this specific domain.

2 Related Work

Recently, several studies have applied continual learning to automatic speech recognition. (Sadhu and Hermansky, 2020) sequentially trained an HMM-DNN model on four different tasks using the Wall Street Journal, Reverb, Librispeech , and Chime4 corpora. (Chang et al., 2021) developed an end-to-end ASR model in which they first pre-trained it on the WSJ corpus then on Librispeech and finally on the Switchboard corpus (Godfrey et al., 1992) tested the model's performance on different speech recognition tasks after each update. As real-world data distributions vary a lot from one task to another, it becomes quite essential to know when the training data is presented with a different task than the one it was trained for. In this respect, (Zeno et al., 2019) came up with a Bayesian approach to continual learning that does not require knowledge at the time of transition from one task to another. Similarly, (Mai et al., 2021) introduced the concept of online continual learning over image classification, assuming that the emergence of new classes or instances of images may include a variety of online data streams.

Applications of continual learning have been successfully extended to various tasks such as computer vision (Aljundi et al., 2017) and automatic speech recognition (Eeckt and hamme, 2023; Fu et al., 2021). This has not been investigated with respect to Speech-to-Text Translation so far.

3 Problem Statement

The continual learning of S2T models is defined as follows. First, we assume we have an initial model that has been trained on a given dataset (D_0). This model serves as a seed model on which a sequence of continual learning updates is applied. Second, we have a set of labeled datasets $D = \{D_i\}_{i=1}^{N}$ which become available sequentially over time for model training. N represents the total number of language pairs available to train the model. Retraining the S2T model from scratch each time a new dataset D_n becomes available incurs substantial computational costs. Hence a replay-based continual learning method retains few samples from previous tasks to minimize the L2 distance between gradients of new and old data, thereby preserving past knowledge while learning new tasks:

$$\mathcal{L}_{total} = \mathcal{L}_{new} + \lambda \sum_{i=0}^{n-1} \|g_{new} - g_{old}^{i}\|_2^2 \quad (1)$$

Where, $\mathcal{L}_{new}$ is the loss on current task, g_{new} is the gradient of the loss with respect to the new data, g_{old}^{i} is the gradient of the loss with respect to the samples from the i-th previous task and λ is a regularization parameter that controls the importance of preserving old knowledge.

4 Methodology

The S2T baseline used is a transformer-based encoder-decoder model (Vaswani et al., 2023). The hypothesis is that the model being trained for future tasks be optimized by comparing the gradients of previous tasks. To that end, we aspire to employ the approach originally proposed for visual recognition task handling continual learning using gradient episodic memory (GEM) (Lopez-Paz and Ranzato, 2022). Motivated by its recent application in computer vision tasks, we ask the following research question: Will the same approach be able to learn in an S2T setting? We confirm that using GEM in S2T setting, we are able to mitigate the catastrophic forgetting.

4.1 Gradient Episodic Memory (GEM)

GEM is a replay-based continual learning method that retains samples from past data in its memory. When the model encounters data from a new task, it minimizes the L2 distance between the gradients of the new data and the old data. To minimize the total loss $\mathcal{L}_{\text{total}}$, we ensure that the update to the model parameters does not significantly change the gradients computed for the old tasks. This constraint helps to prevent catastrophic forgetting. The new gradient is calculated as:

$$g_{\text{new}} = \nabla_\theta \mathcal{L}_{\text{new}} \quad (2)$$

where θ represents the model parameters, and $\mathcal{L}_{\text{new}}$ is the loss function for the new task. The gradients from the stored examples are then calculated as:

$$g_{\text{old}}^{i} = \nabla_\theta \mathcal{L}_{\text{old}}^{i} \quad \text{for} \quad i = 0, \ldots, n-1 \quad (3)$$

where $\mathcal{L}_{\text{old}}^{i}$ is the loss for the samples from the i-th previous task stored in the episodic memory. The gradient ω to prevent catastrophic forgetting is then defined as:

$$\omega = G^\top v + g_{\text{new}} \quad (4)$$

where $G = (g^1_{\text{old}}, \ldots, g^{n-1}_{\text{old}})$ is the matrix of gradients for the previous tasks, and v is a vector obtained by solving the quadratic programming problem that ensures the constraints on gradient alignment.

4.2 S2T Transformer

The Transformer model is an adaptation of the Transformer architecture, specifically designed to handle speech representations as input. These features are inputted into the S2T encoder, which is composed of several layers utilizing self-attention mechanisms. These mechanisms allow the model to process various segments of the input sequence, thereby efficiently capturing long-range dependencies. The self-attention mechanism calculates attention weights to emphasize key features during the decoding process. In the training phase, the model is fine-tuned to align with the ground truth target text by optimizing the following loss function:

$$\mathcal{L}_{ST} = -\sum_{n} \log P(x_n|y_n) \tag{5}$$

Here, $\mathcal{L}_{ST}$ represents the label-smoothed cross-entropy loss on speech and target language text pairs, x is the input speech and y is the target text. This loss is calculated by updating the model parameters θ such that it doesn't change the gradients of previous tasks g_{old}. The S2T Transformer generates a sequence of predicted tokens that articulate the translated textual representation.

5 Experiment

In this section, we detail the following components: (a) datasets, (b) baselines, (c) training and testbed and (d) evaluation metrics.

5.1 Dataset

We conduct experiments on four pairs of translation directions available in **MuST-C**[1] (Di Gangi et al., 2019) dataset: English (En) to German (De), French (Fr), Russian (Ru) and Dutch (Nl). It contains audio, transcript and translation from TED talks for each direction. The statistics of the dataset is shown in Table 1.

MuST-C Dataset					
En →	Hours	#Sents	Train	Val	Test
De	408	274K	269K	1.5K	2.8K
Fr	492	280K	275K	1.4K	2.6K
Ru	489	270K	265K	1.3K	2.5K
Nl	442	253K	248K	1.4K	2.6K

Table 1: Train, test and validation splits for MuST-C.

5.2 Baselines

As there is no previous continual learning baseline for S2T models, we create two baselines of our own. First is to simply *fine-tune* the model on new tasks and the second baseline is a setup where all task's datasets are available together during training as it is a *joint* approach. In this experiment, we consider the *fine-tune* to be a lower bound and *joint* to be an upper bound for the performance of the model.

5.3 Training and Testbed

In this study, we utilized the FAIRSEQ S2T toolkit (Wang et al., 2020) to implement our method. The core architecture is an S2T Transformer encoder-decoder model. Both the encoder and decoder consist of 6 self-attention layers, each featuring 8 attention heads. Due to limitations in training resources, the encoder and decoder are of the *small* configuration, comprising of 256 hidden units. Data augmentation is performed using SpecAugment (Park et al., 2019), and the GELU activation function is employed to enhance convergence, normalization and training stability. The S2T model is trained with label-smoothed cross-entropy loss, with a label smoothing factor set at 0.1. The Adam optimizer is used, featuring a learning rate of 1e-4, and the learning rate schedule follows an inverse square root pattern.

5.4 Performance Metric

Case-sensitive detokenized BLEU using sacreBLEU (Post, 2018) is used to report the performance of the model. All experiments are repeated with three different random seeds, and we report the average BLEU on the MuST-C `tst-COMMON` set.

6 Results

We measure the performance of the system across four tasks sequentially as shown on Table 2. T-1 was conducted on De, T-2 on Fr, T-3 on Ru, and T-4 on Nl. The goal is to retain model performance on previous tasks while performing T-2, T-3 and T-4. The results are given in terms of BLEU scores

---

[1] We use v1.0. `https://ict.fbk.eu/must-c/`

	T-1	**T-2**			**T-3**				**T-4**					Agg.
	De	De	Fr	Avg	De	Fr	Ru	Avg	De	Fr	Ru	Nl	Avg	Avg
Fine-Tune	23.85	0.5	30.1	15.3	0.2	0.2	17.23	5.87	0.3	0.2	0.1	28.78	7.34	7.37
Forg		98%			99.2%	99.3%			98.8%	99.3%	99.4%			98.8%
Joint	26.02	26.02	36.05	31.03	26.02	36.05	18.23	26.76	26.02	36.05	18.23	29.78	27.52	27.84
GEM	23.85	5.42	26.71	16.07	5.59	6.33	14.41	8.78	4.88	5.77	4.32	24.86	9.96	**10.65**
Forg		77%			76.5%	76.3%			79.5%	78.4%	70%			**74.3%**

Table 2: Task-wise average BLEU score and forgetting on four pairs of MuST-C data. Fine-tune and Joint are the baselines whereas GEM is the proposed method for continually learning S2T models. Forg denotes the forgetting on that method. Here, T-1, T-2, T-3, and T-4 are tasks where we train the model on De, Fr, Ru and Nl language pairs sequentially. Agg Avg is overall average.

and Forgetting in percentages, which quantify the retention of tasks learned before.

6.1 Automatic Evaluation

As seen in Table 2 for the Task 1 with $fine\text{-}tune$, the BLEU score for De is 23.85 whereas it significantly lowered in next subsequent tasks. In Task 2, $fine\text{-}tune's$ BLEU score on De lowers to 0.5 and further goes even worse down to 0.2 after Task 3 and after Task 4 to 0.3. It shows similar result with other language pairs as the number of task increases. Conversely, GEM demonstrates quite smooth performance with BLEU score of 23.85 for Task 1 on De, and a score of 5.42 for De after Task 2, 6.33 after Task 3, and 4.32 after Task 4, showing that the model is able to remember the previous task. It follows a similar score for other languages as well. In Figure 2, although Nl in Task 4 is trained for the first time in both $fine\text{-}tune$ and GEM, the increase in BLEU score can be explained due to the forward transfer experienced by the S2T model using GEM. The result shows that GEM is able to preserve previous knowledge at an average BLEU score of 10.65 across all tasks compared to the baseline $fine\text{-}tune$ with an average of 7.37.

Forgetting: One of the main challenges of continual learning is forgetting, which means that the performance on the tasks learned earlier in the run deteriorates upon the introduction of new tasks. From Table 2 we see $fine\text{-}tune$ baseline has very high forgetting rates of 98% on Task 1, 99.2% on Task 2, 98.8% on Task 3, and 99.4% on Task 4. However, this effect of forgetting is considerably reduced if applied GEM: 77% for Task 1, 76.5% for Task 2, 76.3% for Task 3, and 70% for Task 4. This reduces the average rate of forgetting for GEM to be 74.3%.

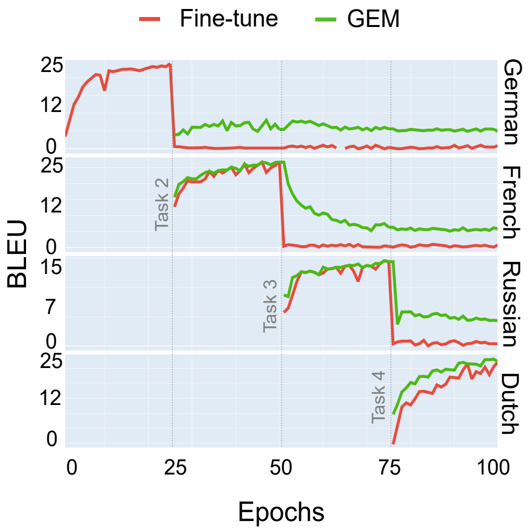


Figure 2: Epoch wise BLEU score on all tasks trained sequentially.

7 Conclusion

In this paper, we propose GemST, a new method for continually learning Speech-to-Text models. Results obtained from our experiments on the MuST-C dataset indicate that GEM not only improves the BLEU scores of multiple tasks compared to the baseline, but it also causes a requisite massive drop in the forgetting rates. Hence it demonstrate GEM's efficacy toward the development of robust S2T systems that learn tasks introduced sequentially without suffering from the so-called catastrophic forgetting. This development paves the way for future research and development on continual learning methodologies within the S2T domain.

Limitations

While our proposed method demonstrates superior performance compared to the baseline, a few limitations should be noted: (1) While GEM effectively retains knowledge from previous tasks, there is potential to further minimize the forgetting. Developing more advanced methods could lead to greater reductions in forgetting, enhancing the overall performance of the model, (2) As this study presents the first application of continual learning to S2T, there is a lack of established baselines. Future work could develop and compare additional continual learning baselines to provide a more comprehensive evaluation. Nevertheless, our primary objective was to initiate research in this area.

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts.

Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. 2017. Towards speech-to-text translation without speech recognition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 474–479, Valencia, Spain. Association for Computational Linguistics.

Andrew Bremner, David Lewkowicz, and Charles Spence. 2013. Multisensory development.

Heng-Jui Chang, Hung yi Lee, and Lin shan Lee. 2021. Towards lifelong learning of end-to-end asr.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Vander Eeckt and Hugo Van hamme. 2023. Continual learning for monolingual end-to-end automatic speech recognition.

Li Fu, Xiaoxiao Li, Libo Zi, Zhengchen Zhang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Incremental learning for end-to-end automatic speech recognition.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 817–824, Online. Association for Computational Linguistics.

David Lopez-Paz and Marc'Aurelio Ranzato. 2022. Gradient episodic memory for continual learning.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. 2021. Online continual learning in image classification: An empirical survey.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, interspeech$_2019.ISCA$.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2022. Progressive neural networks.

Samik Sadhu and Hynek Hermansky. 2020. Continual Learning in Automatic Speech Recognition. In *Proc. Interspeech 2020*, pages 1246–1250.

Balaram Sarkar, Chandresh K Maurya, and Anshuman Agrahri. 2023. Direct speech to text translation: Bridging the modality gap using SimSiam. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 250–255, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. 2019. Task agnostic continual learning using online variational bayes.

CASCA: Leveraging Role-Based Lexical Cues for Robust Multimodal Speaker Diarization via Large Language Models

William Nehrboss
`wnehrboss@casca.ai`

Abstract

In this paper, we introduce CASCA[1] a multimodal speech diarization framework that incorporates speaker role information. Motivated by the challenges of diarizing single-source customer-employee interactions in noisy environments, this framework utilizes a cascading sequence of fine-tuned large language models to characterize distinctions in speaker roles. Audio with linguistic content associated with particular roles is used to formulate acoustic speaker profiles; these profiles reduce the subsequent clustering task into a classification task. CASCA is robust to sparsity or low signal-to-noise ratios, conditions that tend to confound traditional clustering algorithms. Although intended for those domains with clear role distinction, e.g., doctor-patient, teller-customer, through topic segmentation, CASCA captures transient, topic-level speaker role information to reliably identify speaker profiles. This expands the domain of applicability. We validate the effectiveness of our approach on a benchmark of two-speaker conversations from a variety of domains, achieving an 80% reduction in word diarization error rates over our conventional baseline.

1 Introduction

Speaker diarization is the process of segmenting recorded audio according to the speaker source. It determines who spoke when by splicing audio into regions of homogeneous speaker source and applying a speaker tag to those regions. Accurate speaker diarization is crucial for effective conversation understanding, which is essential in myriad applications from customer service analytics to medical recordkeeping. Spoken conversations are rich in both linguistic and acoustic information. However, most current diarization systems utilize only acoustic information in speaker assignment. Some of the most popular diarization algorithms (Serafini et al., 2023), including Pyannote (Plaquet and Bredin, 2023), which we use as our baseline, are cluster-based. The general architecture of these systems is as follows:

- *Voice activity detection* isolates speech from non-speech.
- *Segmentation* splits regions of speech into smaller segments with a single active speaker.
- *Embedding extraction* yields vector representations capturing key audio characteristics.
- *Clustering* groups these embeddings to determine speaker assignment.

These systems, however, tend to generalize poorly to varied real-world situations. Embedding clusters are often imbalanced, non-Gaussian, or indistinct due to uneven speaker participation, shifts in tone or intonation, or background noise. These factors make accurately identifying cluster boundaries, and, in turn, speaker assignment unreliable. A prime example of a conversation that yields indistinct embedding clusters is presented in Table 3a. This work aims to solve these challenges by reformulating the clustering step into a classification step by incorporating a key source of speaker differentiation available in the linguistic content of the conversation: speaker roles.

Speaker roles within a given conversation tend to be distinct. The degree of this distinction can be high, for example, in conversations between a doctor and patient or salesperson and customer, or low, such as in casual conversations between two siblings or two roommates. Speaker roles can provide strong cues about the correct speaker assignment of certain speech segments within a conversation. For example, in a conversation between a doctor and a patient, the speech segments corresponding to the phrases “I am experiencing chest pain” and “I am going to recommend an X-ray” can be at-

---

[1] **C**ontext **A**ware **S**peech **C**lassification **A**rchitecture `https://github.com/CASCA-Labs/CASCA`

tributed to the roles of the patient and the doctor, respectively. This information can be used to isolate particular speech segments that correspond to each role. These segments, representing a sort of acoustic speech profile of each speaker, simplify the subsequent speaker assignment task into a classification task, avoiding the need for unreliable clustering algorithms.

1.1 Types of Speaker Role Distinction

In this paper, we will refer to two types of speaker role distinction: strong role distinction and weak role distinction. Strong role distinction is present when the role of each speaker is stable and pertains to the speaker themselves, more or less independent of the conversation. For example, in the case of a conversation between a bank teller and a customer, the roles of the speakers and what they might be expected to say are strongly determined by their relationship to the service being provided. Weak role distinction is present when the role of each speaker is fluid throughout the conversation. In these cases, there are no overriding contextual factors that explain the linguistic content. Importantly, however, even when roles are more fluid, speakers typically assume identifiable roles within certain segments of the conversation that relate to specific topics. For instance, in the conversation summary presented in Table 1, although the speakers are peers without apparent strong role distinction, they assume different roles within each topic segment: one informs the other about a promotion in the first segment and updates her about a mutual friend in the second. Leveraging this weak role distinction presents a challenge but is crucial to the robustness of our approach. This motivates the specification of the first stage of our pipeline, which extracts role distinctions on the topic level (see Sections 2.1–2.3).

1.2 Prior Work

Utilizing linguistic information is recognized as a key opportunity to enhance diarization systems. Recent advances in n-gram models, particularly transformer-based models, have made the use of this information more accessible and valuable. Multimodal diarization approaches leveraging these models have proven effective. BERT-based models, for instance, have shown promise in post-processing transcribed dialogues and correcting errors from misaligned speaker turns (Paturi et al., 2023). Efforts have also been made to use a priori knowledge of speaker identities for downstream classification tasks in different contexts. One study (Flemotomos et al., 2020) involved training classifiers on sentence-level speech segments to construct speaker profiles in therapist-patient conversations. A subsequent investigation (Flemotomos and Narayanan, 2022) extended this approach to two domains, using linguistic information to constrain embedding clusters. Another study (Prasad et al., 2021) addressed problematic audio data in an aviation setting using a related method. Although these efforts are valuable, they are limited in certain respects, particularly in their dependence upon a priori information of speaker identities. In this paper, we tackle the more difficult problem of role-aided diarization without prior knowledge of speaker identities, where leveraging relational information is a central aspect of our methodology.

2 LLM Sequence

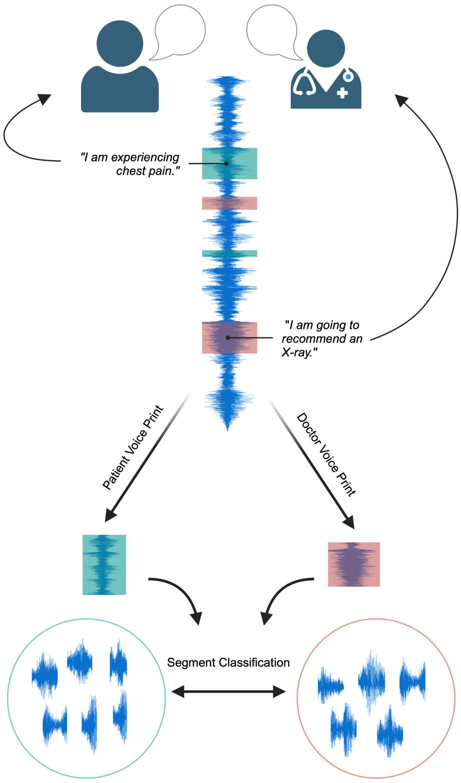

"I am experiencing chest pain."
"I am going to recommend an X-ray."
Patient Voice Print
Doctor Voice Print
Segment Classification

Figure 1: The motivation for CASCA. The roles of the doctor and patient can be used identify segments of speech that belong to each speaker. These segments compose acoustic speaker profiles against which can be used in speaker assignment of the remaining segments.

Characterizing speaker roles serves as the foundational step in isolating speech profiles. Firstly, transcripts are generated through an ASR model, which are then passed to three specialist LLMs: a summarization model, a topic segmentation model, and a role identification model. As each model processes a smaller segment of the conversation, information is passed downstream at each stage, allowing the maintenance of high-level context throughout. After roles are identified, a fourth specialist LLM identifies those segments of transcribed speech most likely to be associated with each role. The corresponding speech segments are then combined to form speech profiles for each speaker. Vector embeddings are generated for each speech profile as well as each speech segment; speech segments are then assigned a speaker source according to the maximum cosine similarity to the corresponding speech profiles. We highlight two conversations from our experiment: CALLHOME 0638 (see Table 1) from the CALLHOME (Canavan et al., 1997) dataset and MedData RES0102 (see Table 2) from the MedData (Farzandipour et al., 2022) dataset. CALLHOME 0638 is an example of a conversation with weak role distinction, elucidating the need for the topic segmentation stage. MedData RES0102 is an example of a conversation with fuzzy embeddings cluster boundaries (see Figure 3a) that cannot be accurately diarized using audio alone. Using role information, speech segments attributable to each speaker are used to build acoustic speech profiles (see Figure 3b), facilitating accurate speaker assignment (see Figure 4).

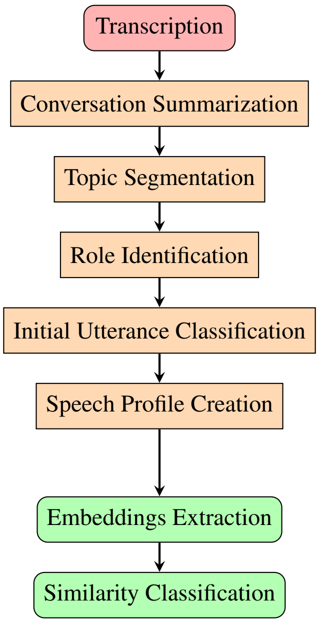


Figure 2: Architecture of CASCA

2.1 Summarization

The first model in our LLM pipeline summarizes the nature and type of the conversation, such as an interview, transaction, or friendly chat. This summary is passed downstream in the prompt of the role identification model to maintain context that could be lost during topic segmentation. Preserving the high-level context ensures that the assigned role profiles remain consistent with the nature of the conversation.

2.2 Topic Segmentation

Topic Segmentation of CallHome 0638, Chunk 01.

At Kinko's? | Mm-hmm. | Is it the same job that you had when I left? | Well, I'm almost out of my job. | What does that mean? | They... They gave me a promotion to supervisor. | Wow. | Yeah, but I can't change out of it until the 10th of August. | Oh. | But I've been working on Sundays as a supervisor. | So I've sort of been the supervisor, but I still have my other job during the week. | Do you like being supervisor? | Yeah, I do. | Oh, good. | Okay. | Well, that's good. | And then I can work, like, evenings during school. | Uh-huh. | I don't have to work in the day. | Oh, very good. And how's Leroy? | He's okay. | He moved into Felipe's apartment. | Uh-huh. | And it was about, what, a week ago? | Uh-huh. | And I haven't seen much of him since. | See, I heard that they split up again.
Model Response: First Theme: Career Transition and Job Promotion. First Line: Just working. | Line That Transitions Away From Theme: And how's Leroy? Second Theme: Personal Life and Leroy's Move. First Line: And then I can work, like, evenings during school. | Line That Transitions Away From Theme: See, I heard that they split up again.

Table 1: This conversation lacks strong role distinction. The speakers are female friends, and the role each plays is fluid. However, their roles are distinct and stable within each topic segment. Topic 1: one speaker informs the other about a promotion in the first segment Topic 2: one speaker updates the other about a mutual friend.

The second model splits the conversation into regions with a homogeneous topic to facilitate the extraction of topic-level role information. This allows for the use of role information in conversations with weak role distinction (see Table 1). To do so, lines marking conversational boundaries are identified. Topic segmentation results in a list of conversation segments each pertaining to the same conversational object; the downstream role and line assignment models subsequently work with these thematically unified segments to identify and apply the role information.

2.3 Identification of Speaker Roles

Utterance	Speaker	Conf.
So yeah, it's the cough.	01	0.2
It started a couple of weeks ago.	02	1.0
A couple of weeks ago?	01	1.0
Okay.	01	0.13
And has it gone worse since then?	01	1.0
Or has it stayed about the same?	01	1.0
It got worse initially, but it's been about the... Well, actually, yeah, it's been getting worse since now.	02	0.95
I've started to... noticed blood in this freedom.	02	.34
I wasn't there at first.	02	0.45
Okay, when did you first notice that?	01	1.0
So I first saw some blood a few days ago.	02	1.0
It was a really small amount, so I didn't really see much, but I brought out blood.	02	0.95
Yesterday, and again this morning, it's been just about enough to cover 50, so it's not a lot of driving time, but it's pretty rough.	02	0.95
Okay, and before that, were you getting any production for your cult for the last few years?	01	0.15
Like, were you producing any music?	01	0.23
Uh, no.	02	0.3
No?	01	0.5
Okay.	01	0.1
Can you just describe your cough term?	01	1.0
Is it a wet cough or dry cough?	01	0.95
It's dry, but it's really with the exception of the blood.	02	0.95

Table 2: *MedData, Conversation RES0102* In this example, the two identified speaker roles, doctor and patient, are used to positively identify certain segments as belonging to each speaker.

The third model determines the roles of each speaker in a specific topic chunk, in the context of the broader conversation summary. The two following examples illustrate how speaker roles are found in cases of both strong and weak role distinction.

2.3.1 Weak Role Distinction

In the absence of clearly distinct, stable roles, the model relies on the differences in each speaker's relationship to the central topic of the conversation within the topic segment to define speaker roles. For instance, in the first topic segment of CallHome 0638 Table 1, the two speakers are distinct in their roles as informant and informee.

Model Response: *SPEAKER A is sharing updates about their job change and the current situation, seeking validation and support from SPEAKER B. SPEAKER B's role is that of a listener and supporter [showing] interest in SPEAKER A's well-being and the benefits of the job change, such as having more flexibility in working hours.*

2.3.2 Strong Role Distinction

The model can more easily extract role information when consistent distinctions are present. These are constrained by the conversation context established by the first model. For example, the model accurately characterizes the roles of the patient and doctor in MedData RES0102.

Model Response: *Speaker_01, who is sharing their symptoms with Speaker_02, who is likely a medical professional or seeking to understand Speaker_01's symptoms in a medical context. Speaker_01 is the individual experiencing and reporting their symptoms.*

2.4 Speech Profile Creation

Each transcribed utterance is passed to a fifth model, along with the surrounding conversational context and speaker roles identified in the previous step for classification. We assign the logarithmic probability (logprob) associated with the speaker label token as the confidence score for the classification. We explored several alternatives for this confidence score, including repeated prompt agreement (Portillo Wightman et al., 2023) and auxiliary fine-tuned models to determine confidence, but found these approaches to be too computationally expensive or unreliable. Utterances that are clearly associated with a particular role - to use the same example, "I am going to recommend an X-ray" which is clearly associated with the role of a doctor - tend to be classified with greater confidence. We then take the set of utterances with the highest confidence scores to form our speech profiles corresponding to the respective roles. The detailed algorithms used to mix these utterances are found in Section A.

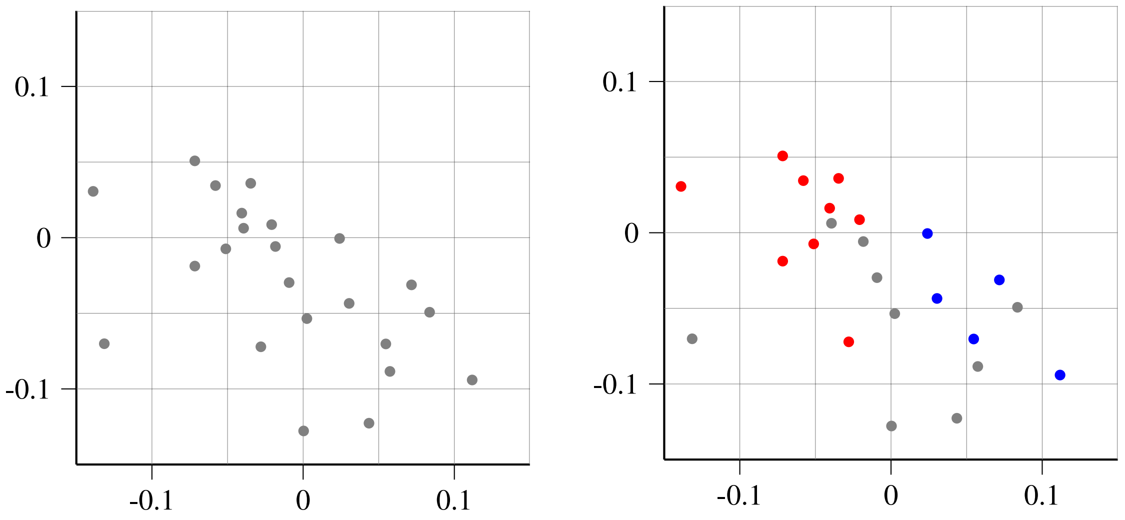
2.5 Final Classification

The vector embedding of each segment, σ_i, is calculated, and each segment is then classified according to similarity to each speech profile, α_j: $\max_j \in speakersetsim(\sigma_{x_i}, \alpha_j)$.

3 Experiment

3.1 Models

In the first step of automatic speech recognition (ASR), we use the Whisper Large V3 model with word-level alignment and segmentation using WhisperX (Radford et al., 2023; Bain et al., 2023). The tendency of Whisper to remove disfluencies, i.e. *"I I don't"*, *"uh"*, etc., significantly increased word error rates (WER) on verbatim transcripts. We chose WhisperX due to the reliability of generated timestamps. Transcripts are broken into utterances: each identified utterance is almost



(a) Cross cosine similarity of utterance embeddings MedData RES0102.

(b) Utterances labelled as Speaker 1 or Speaker 2 according to identified speaker roles.

Figure 3: This example highlights how role-aware diarization succeeds where traditional acoustic methods fail. The noisy audio of a short interaction results in embeddings with no identifiable clusters (Fig. 3a). However, the previously identified speaker roles of the doctor and patient inform the assignment of some of the utterances (Fig. 3b) to each speaker. This clarifies the acoustic distinctions between speakers. The subsequent speaker assignments using the speech profiles (see Figure 4) are nearly perfect.

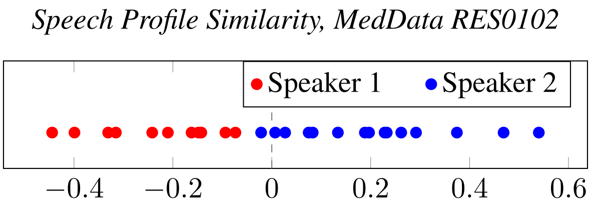


Figure 4: Difference in cosine similarity of each speech segment to the acoustic speaker profile of the patient and doctor respectively. The high accuracy of this classification far outperforms the audio-only clustering methods. Note: The one misclassified segment is Segment #19 (text: 'No', duration: 0.08 seconds) is extremely short; embeddings become unstable as speech segment length decreases.

always from a single spoken speaker. However, utterances are not separated by speaker turn; consecutive utterances may or may not be from the same speaker.

Each component model of our LLM sequence is a task-specific fine-tune of Mistral7B-Instruct-V1 (Jiang et al., 2023; Wang et al., 2024a). We chose this base model for a few reasons. Firstly, it is open source. Secondly, its small size lightens the computation burden of repeated LLM calls that the framework entails. Thirdly, it helps illustrate the potential of effectively fine-tuned, specialist small language models in diarization, a secondary contribution of this work. Current speech processing projects, for example (Wang et al., 2024b), are built upon extremely large and computationally costly models; CASCA demonstrates that limitations in reasoning associated with lower-parameter models can be overcome through careful delegation of tasks and fine-tuning.

We use the WavLM-Large embedding model (Chen et al., 2022) for embedding speech segments and each speech profile.

3.2 Fine-Tuning

3.2.1 Generation of Training Data

The raw material for our fine-tuning data was sourced from open-source datasets of real-world dialogue, including DailyDialogue (Li et al., 2017), SWDA: Switchboard Dialogue Act (Jurafsky et al., 1997), and CallFriend (Canavan and Zipperlen, 1996). We used GPT-4 (OpenAI, 2023) to generate task-appropriate responses corresponding to conversations drawn from these datasets. The data generation methodology was specific to each task, depending on its complexity.

- **Conversation Summarization:** Straightforward, single-step prompting proved sufficient to generate accurate conversation summaries.

- **Topic Segmentation:** We utilized a two-stage chain-of-thought (CoT) prompting approach to assist the model in identifying major topics and subsequently determining the boundary phrases of each topic.

- **Role Identification:** We paired conversation summaries created earlier with conversation transcripts to identify speaker roles, emphasizing distinctions between the roles.

- **Utterance Assignment:** Iterating through each utterance in the conversations, we provided the model with the identified speaker roles, the target utterance, the surrounding context, and the correct utterance label. We prompted the model to explain the logical progression from the provided information to the correct answer. This produced a data set that provided effective logical instruction for this task, as the base model initially performed poorly on this type of reasoning-based task.

Task	Source	Entries	Method
Summarization	DD, SWDA	~2000	Few-Shot
Topic Segmentation	CallFriend	~500	CoT
Role Extraction	DD, SWDA	~2000	Few-Shot
Line Assignment	DD, SWDA	~3000	CoT

Table 3: DD: DailyDialogue; SWDA: Switchboard Dialogue Act; CallFriend

We fine-tuned using a LoRA adapter with a learning rate of 0.0002 and cosine decay. Training lasted for three epochs using 4-bit quantization for computational optimization. Data preprocessing included a random split (95%) - (5%) training-validation. The fine-tuning data is made publicly available. [2]

3.3 Test Data

In this section, we present the results of our approach on 96 hours of out-of-domain conversation data from various settings, collected mainly from TalkBank (MacWhinney, 2023). We constructed our evaluation set from selected subsets of two-speaker conversations chosen from available TalkBank data, without any prior knowledge of the audio. These data sets include CORAAL (Kendall and Farrington, 2023), featuring interviews with African-American participants; CALLHOME (Canavan et al., 1997), comprising telephonic conversations between friends and family; and an open-source set of simulated doctor-patient conversations (Farzandipour et al., 2022), which we mix with background noise to simulate challenging real-world conditions[3]. A few conversations from two miscellaneous sources, MICASE (Simpson et al., 2002), containing academic dialogue, and SBCBASE (Du Bois et al., 2000–2005), containing a mix of informal dialogue, were also included to explore different conversation scenarios. Selections from MICASE and SBCBASE were limited to the few two-speaker conversations available in these datasets. This experiment took about 4 hours of active GPU usage on an A100.

3.4 Evaluation Metrics

Unlike most diarization systems, ASR transcription is an integral part of our framework. CASCA is oriented towards the classification of already transcribed speech segments; therefore, we can use the word error rate to measure the accuracy of our system. The fidelity of the final transcripts effectively conveys how well conversational information is preserved during the entire pipeline of speech processing. Our metric of interest is Word Diarization Error Rate (**WDER**), which is used to evaluate diarization systems that incorporate ASR (Shafey et al., 2019; Tran et al., 2022). To define WDER, we first specify its two component metrics, Word Error Rate (**WER**) and Concatenated Permutation Word Error Rate (**cpWER**).

WER captures the accuracy of the transcription without considering the speaker identification error. It is calculated as:

$$WER = \frac{lev(R, H)}{|R|} \tag{1}$$

cpWER for two speakers accounts errors from both transcription and incorrect speaker speaker attributions (diarization errors). Given that hypothesis H_1 corresponds directly to reference R_1, and hypothesis H_2 to reference R_2, the cpWER is calculated as:

$$cpWER = \frac{lev(R_1, H_1) + lev(R_2, H_2)}{|R_1| + |R_2|} \tag{2}$$

$lev(R, H)$ represents the Levenshtein distance between the reference transcript R and the hypothesis transcript H, and $|R|$ is the total number of words in the reference transcript. Finally, WDER is specified as:

$$\textbf{WDER = cpWER - WER} \tag{3}$$

---

[2] `https://github.com/CASCA-Labs/CASCA`

[3] Background noise is mixed to achieve a signal-to-noise ratio of 11.

Source	# Dialogues	Total (m)	# Words	# Segs	Avg Seg Length (s)	Avg Words/Seg
CALLHOME	90	1795.10	289785	23277	4.6	12.45
MedData	266	3138.56	620210	50195	3.7	12.35
CORAAL	23	628.78	176037	14129	2.7	12.46
MISC	7	224.80	68072	6012	2.2	11.32

Table 4: Composition of Evaluation Set

This is less forgiving than other specifications of WDER or the analogous time-based Diarization Error Rate (DER). Under this specification, confused speaker assignments are counted as both an insertion in the stream where they are erroneously added and a deletion in the stream from which they are missing.

We calculate these metrics using the MeetEval toolkit (von Neumann et al., 2023). Technically, we employ a time-constrained optimal reference combination word error rate to calculate WER and a time-constrained concatenated minimum permutation word error rate to cpWER. Time constraints reduce the computational burden and result in only a negligible overestimate of the true cpWER.

3.5 Baseline

To contextualize the marginal value of role distinction in diarization, we present a baseline audio-only diarization system. For this purpose, we employ Pyannote (Plaquet and Bredin, 2023), which is integrated into the WhisperX framework. Pyannote is one of the most popular diarization frameworks and achieves competitive performance on most diarization tasks. Its integration with WhisperX is advantageous as it enables an equitable comparison of the two methods, each utilizing the same ASR output and attempting to classify speaker segments bounded by the same timestamps.

3.6 Results

3.6.1 ASR

Whisper-V3 yields an ASR error rate of 16.6% across all conversations. The ground truth transcripts are verbatim transcripts, which contain disfluencies, nonstandard nomenclature, or names; this is the source of much of the ASR error. This error rate is in line with benchmarks for the model; our reported CALLHOME ASR word error rate of 19.75% is within 2% of the standard achieved in OpenAI's technical report (Radford et al., 2023). This difference is partially or wholly explained by less robust word standardization. Note that more linguistic information is retained than this error rate suggests, as incorrect transcription of disfluencies tends to have little impact on meaning.

3.6.2 Baseline Performance

Our baseline achieves a mean WDER error rate of 22%. The distribution of errors is somewhat bimodal (see Figure 5). This is due to the presence of conversations in which the differentiation in the acoustic characteristics of each speaker's voice is insufficient to clearly define clusters in the utterance embeddings (e.g., Figure 1). This causes the clustering algorithm to go awry and, in turn, results in extremely inaccurate diarization.

3.6.3 CASCA

CASCA exhibits markedly improved performance across the evaluation set. The mean WDER of 4.2% represents an 80% improvement in accuracy. As expected, CASCA performs best in the presence of strong role distinction, such as in the professional MedData conversations, and worst in the presence of weak role distinction, such as in the casual CALLHOME conversations. However, even in those cases, CASCA still outperforms the baseline, a result that validates the utility of the topic segmentation stage.

4 Conclusions and Future Work

In this paper, we offer a conceptual framework for the dynamic utilization of speaker role distinction in speaker diarization through a sequence of specialized LLMs. We demonstrate that this framework far surpasses acoustic only diarizaiton for a variety of conversation types. Performance varies with the strength of the distinction between speaker roles. These results highlight the potential of leveraging the rich role information contained within the conversational text for the task of speaker diarization. Tracking our original motivation for this project, our expectation is that this framework will offer the most value in physical commercial settings, where speaker roles are very distinct but hostile recording environments make

Source	ASR WER	Baseline WDER	CASCA WDER	Improvement	Role Distinction
CALLHOME	19.75%	38.8%	13.5%	25.3%	Weak
MedData	14.44%	16.1%	1.6%	14.5%	Strong
CORAAL	22.53%	48.6%	7.3%	41.3%	Strong
MISC[4]	24.18%	56.4%	4.7%	51.7%	Moderate

Table 5: CASCA achieves a 4.2% average WDER, outperforming the baseline of 22%.

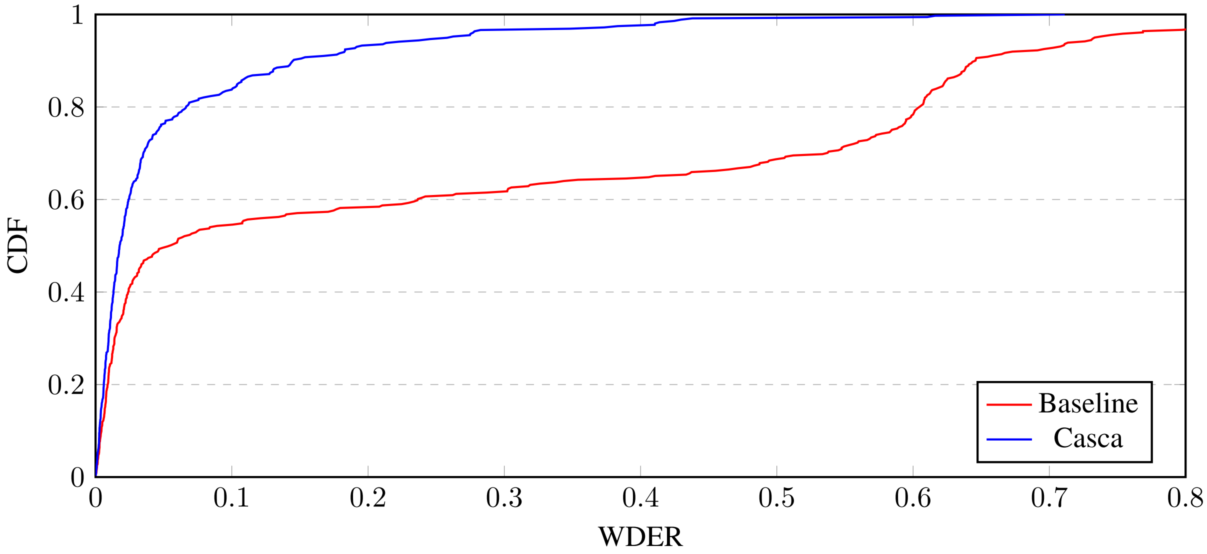


Figure 5: Comparison of cumulative distribution functions of errors for both CASCA and baseline. CASCA displays higher accuracy, especially in cases where the vocal characteristics of participant voices are similar.

acoustic clustering challenging. Our use of seven billion parameter LLMs is also notable as it reduces the cost of the system while illustrating the potential for downsizing speech processing models to fine-tuned specialists. An attractive next step of research is to explore other methods of utilizing the identified speech segments in speaker assignment. Our current method of classifying speech segments according to cosine-similarity speech profiles is simplistic. Other methods, such as the use of reference segments to constrain relationships between embeddings in the definition of clusters, could be more reliable. Additionally, the linguistic content of the speech profiles could be used to develop more sophisticated speaker profiles by identifying speaker's pronunciation of particular words. This would further simplify speaker assignment into a type of text-dependent speaker verification.

5 Limitations

A significant limitation of this study is its exclusive focus on dyadic conversations. In two-speaker interactions, role distinctions are generally apparent and informative. However, with additional participants, these distinctions become increasingly ambiguous. Discerning unique roles in multiparty conversations without prior information is exceptionally challenging, barring specific contextual factors such as commercial interactions where participants have distinct relationships to the subject matter. One potential approach for multiparty conversations could involve progressively identifying roles - establishing one speaker's role, using that context to inform another, and iteratively uncovering roles until the set is fully specified. The feasibility of this method, along with alternative approaches for extending this framework to conversations with more participants, remains a topic for future research. Another limitation stems from the framework's reliance on the accuracy of initial ASR transcription. If sufficiently severe as to affect meaning, errors in this stage could confound the downstream role analysis, undermining the entire diarization process. Finally, the computational demands of sequential specialized LLM processing present a practical limitation. Although the use of smaller language models mitigates this issue to some extent, the computational cost still substantially exceeds that of audio-based diarization systems. Current audio-based systems can achieve processing speeds exceeding 60 times real-time, whereas our system averages only 5 times real-time using an A100 GPU.

References

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. WhisperX: Time-accurate speech transcription of long-form audio. In *Proceedings of INTERSPEECH 2023*.

Alexandra Canavan, David Graff, and George Zipperlen. 1997. CALLHOME american english speech (LDC97S42).

Alexandra Canavan and George Zipperlen. 1996. CALLFRIEND american english–non-southern dialect (LDC96S46).

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

John W. Du Bois, Wallace L. Chafe, Charles Meyer, Sandra A. Thompson, Robert Englebretson, and Nii Martey. 2000–2005. Santa barbara corpus of spoken american english, parts 1–4.

Fareez Farzandipour, Tapan Parikh, Charles Wavell, and et al. 2022. A dataset of simulated patient-physician medical interviews with a focus on respiratory cases. *Scientific Data*, 9:313.

Nikolaos Flemotomos, Panayiotis Georgiou, and Shrikanth Narayanan. 2020. Linguistically aided speaker diarization using speaker role information. In *Proceedings of Odyssey 2020: The Speaker and Language Recognition Workshop*, pages 119–126. ISCA.

Nikolaos Flemotomos and Shrikanth Narayanan. 2022. Multimodal clustering with role induced constraints for speaker diarization. In *Proceedings of INTERSPEECH 2022*, pages 3518–3522.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Dan Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado at Boulder and SRI International.

Tyler Kendall and Charlie Farrington. 2023. The corpus of regional african american language. Version 2023.06.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Brian MacWhinney. 2023. Talkbank. Accessed: 2023-06-16.

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Rohit Paturi, Sundararajan Srinivasan, and Xiang Li. 2023. Lexical speaker error correction: Leveraging language models for speaker diarization error correction. In *Proceedings of INTERSPEECH 2023*.

Alexis Plaquet and Hervé Bredin. 2023. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proceedings of INTERSPEECH 2023*.

Arnab Poddar, Md Sahidullah, and Goutam Saha. 2018. Speaker verification with short utterances: A review of challenges, trends and opportunities. *IET Biometrics*, 7(2):91–101.

Gwenyth Portillo Wightman, Alexandra DeLucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–332, Toronto, Canada. Association for Computational Linguistics.

Amrutha Prasad, Juan Zuluaga-Gomez, Petr Motlicek, Saeed Sarfjoo, Iuliia Nigmatulina, Oliver Ohneiser, and Hartmut Helmke. 2021. Grammar based speaker role identification for air traffic control speech recognition. *Preprint*, arXiv:2108.12175.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR.

Luca Serafini, Samuele Cornell, Giovanni Morrone, Enrico Zovato, Alessio Brutti, and Stefano Squartini. 2023. An experimental review of speaker diarization methods with application to two-speaker conversational telephone speech recordings. *Computer Speech Language*, 82:101534.

Laurent Shafey, Hagen Soltau, and Izhak Shafran. 2019. Joint speech recognition and speaker diarization via sequence transduction. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 396–403.

R. C. Simpson, S. L. Briggs, J. Ovens, and J. M. Swales. 2002. The michigan corpus of academic spoken english.

Bao D. Tran, Ramesh Mangu, Ming Tai-Seale, Jennifer E. Lafata, and Kai Zheng. 2022. Automatic speech recognition performance for digital scribes: A performance comparison between general-purpose and specialized models tuned for patient-clinician conversations. *AMIA Annual Symposium Proceedings*, 2022:1072–1080.

Thilo von Neumann, Christoph Boeddeker, Marc Delcroix, and Reinhold Haeb-Umbach. 2023. MeetEval: A toolkit for computation of word error rates for meeting transcription systems. *Preprint*, arXiv:2307.11394.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Quan Wang, Yiling Huang, Guanlong Zhao, Evan Clark, Wei Xia, and Hank Liao. 2024b. DiarizationLM: Speaker diarization post-processing with large language models. *Preprint*, arXiv:2401.03506.

A Algorithms for Creation of Speech Profiles from Utterances

A.1 Speech Profile Mixing

Each assigned utterance with a confidence score above the a threshold τ_c is appended to the topic-level speech profile for the assigned speaker. Because we only need a small fraction of all classified utterances for reference speech profiles, we can tolerate a high rate of false negatives; we use a conservative $\tau_c = 0.99$

The end result is two sets X of segments for each topic, each set containing speech utterances from opposing speakers. The relation between labels within the same topic is known, but the assignment of speaker labels "A" or "B" is arbitrary between topics. Therefore, we mixed these segments using embedding similarity. We clean and mix the topic-level segment sets according to the following algorithms:

A.1.1 Clean Topic Sets

Let $X = \{\sigma_{x_{ij}} \mid i \in topics, j \in speakers\}$ be the set of segments of each topic of opposing speakers. For each pair of segments $\sigma_{x_{ij}}$ and $\sigma_{x_{kl}}$ in X, calculate the cosine similarity:
$sim(\sigma_{x_{ij}}, \sigma_{x_{kl}}) = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|}$ where $\vec{A}$ and $\vec{B}$ are the embedding vectors of $\sigma_{x_{ij}}$ and $\sigma_{x_{kl}}$, respectively.

For each segment $\sigma_{x_{ij}}$, calculate the average cross similarity by averaging the similarities of $\sigma_{x_{ij}}$ with all other segments. The centroid segment x_c is the one with the maximum average similarity for $\sigma_{x_{ij}}$. Retain a segment if its similarity to x_c exceeds $.2 \times \tilde{x}$, where $\tilde{x}$ represents the median similarity to the centroid x_c.

A.1.2 Mix Topic Sets

The resulting homogenized pairs are then mixed according to the combination that maximizes the joint cosine similarity of the mixed pairs. This process is highly reliable due to the length of the audio in each subtopic speech profile; longer speech strings yield more reliable embeddings (Paturi et al., 2023). The richer phonetic information available allows the embedding model to more effectively capture the characteristics of the speaker's voice; indeed, (Poddar et al., 2018) showed that there is a monotonic relationship between the length of the speech segment and the accuracy of the embeddings. This fact makes the successive merging of the topic-level speech profiles highly reliable.

Acknowledgements

Briefly, we would like to extend our gratitude to Max Bain for the excellent timestamp alignment in his WhisperX system. We also thank the MeetEval team for their exceptional work in developing a user-friendly and powerful error measurement system. Without those systems, this project would have been significantly more difficult.

Deep Information Maximisation to Mitigate Information Loss in Text Independent Speaker Verification

Nipun Fonseka, Nirmal Sankalana, Buddhika Karunarathne, Uthayasanker Thayasivam
Dept. of Computer Science & Engineering
University of Moratuwa
Sri Lanka
{nipunf.19, nirmalsankalana.19, buddhika, rtuthaya}@cse.mrt.ac.lk

Abstract

This paper presents a novel approach to mitigate information loss in text-independent speaker verification by leveraging Deep Information Maximisation (DIM). The proposed method aims to enhance the retention of speaker-specific information during the pooling process, which is crucial for creating accurate and high-level speech signal representations. By incorporating mutual information maximisation techniques, the DIM method optimises the statistical dependency between frame-level features and their corresponding high-level embeddings. Experiments conducted on the VoxCeleb1 dataset demonstrate the efficacy of the approach, showing a significant reduction in the Equal Error Rate (EER). Our best configuration achieved an EER of 1.5376, an improvement over the baseline model's EER of 1.6119. These findings indicate that the integration of DIM can effectively enhance the performance and accuracy of speaker verification systems.

1 Introduction

Speaker verification is the task of determining whether a speaker's claimed identity is true. This process involves two main phases: the first is converting a speech signal into a fixed-dimensional, high-level representation called an embedding; the second is measuring the similarity between such embedding to verify identity.

In text-independent speaker verification, pooling is essential for combining frame-level features into a single, higher-level representation. However, this process can lead to the loss of crucial speaker information, vital for accurate speaker verification. Various techniques have been proposed to address this issue, including attention-based pooling (Okabe et al., 2018), multi-level pooling (Tang et al., 2019), and vector-based attentive pooling (Gao et al., 2020). Despite these advancements, significant information can still be lost due to the inherent compression involved in pooling.

With the rise of deep learning, deep neural networks have become widely used in speaker verification for producing consistent, high-level representations of speech signals. Starting with x-vector systems (Snyder et al., 2018), various methods have been developed over time, including Time Delay Neural Networks (TDNNs) (Liu et al., 2022), Long Short-Term Memory networks (LSTMs) (Mobiny and Najarian, 2018), Extended Context-Aware Permutation-Invariant TDNNs (ECAPA-TDNNs) (Desplanques et al., 2020), Convolutional Neural Networks (CNNs) (Zhou et al., 2019; Zhao et al., 2020; Kim et al., 2022), and more recently, transformers (Peng et al., 2023). While these neural networks are effective at feature extraction, the pooling process is crucial for creating fixed-dimensional high-level representations. Researchers have developed several pooling methods, evolving from statistical (Variani et al., 2014) techniques to advanced methods like multi-headed attentive pooling (Zhu et al., 2018), aiming to optimise the pooling process.

Mutual information is a measure that quantifies the statistical dependence between two random variables. Belghazi et al. introduced a neural network-based method called the Mutual Information Neural Estimator (MINE) (Belghazi et al., 2018) for estimating the mutual information between two variables. This method was adapted by Hjelm et al. to create deep representations of images by maximising the mutual information between an image and its high-level representation (embedding) (Hjelm et al., 2019).

In this research, we apply the Deep Information Maximisation technique proposed by Hjelm et al. to mitigate information loss during pooling in speaker verification. Our approach aims to enhance the retention of speaker-specific information, thereby improving the performance and accuracy of text-independent speaker verification systems.

2 Related Work

Our baseline model for speaker verification builds on the work of Peng et al., which introduces an attention-based backend for fine-tuning pre-trained Automatic Speech Recognition (ASR) Transformer models. This approach leverages the ability of pre-trained Transformers to capture meaningful acoustic and phonetic representations while introducing a lightweight backend to extract speaker-discriminative features effectively (Peng et al., 2023).

The core component of the attention-based backend is the Multi-Head Factorised Attentive Pooling (MHFA) mechanism (Peng et al., 2023). It aims to condition the speaker representations on the phonetic content of the input utterance, enabling the model to capture both speaker and phonetic information simultaneously. The output feature map of each layer of the transformer is utilised here by assigning two types of attention weights.

Given the output representations $\mathbf{Z}_l \in R^{T\times F}$ from the l-th Transformer layer of the pre-trained model, where T is the number of frames and F is the feature dimension, the MHFA method computes two factorised representations: keys $\mathbf{K}$ and values $\mathbf{V}$, as follows:

$$\mathbf{K} = \sum_{l=1}^{L} w_k^l \mathbf{Z}_l \mathbf{S}_k, \quad \mathbf{V} = \sum_{l=1}^{L} w_v^l \mathbf{Z}_l \mathbf{S}_v \tag{1}$$

Here, $\mathbf{w}_k^l$ and $\mathbf{w}_v^l$ are learnable weights that aggregate the layer-wise outputs, and $\mathbf{S}_k \in R^{F\times D}$ and $\mathbf{S}_v \in R^{F\times D}$ are linear projections that reduce the dimensionality of keys and values, respectively, to $\mathbf{D}$.

The multi-head attention mechanism is then applied to aggregate the values $\mathbf{V}$ over frames, conditioned on the keys $\mathbf{K}$:

$$\mathbf{A} = \mathrm{softmax}(\mathbf{K}\mathbf{Q}^\top) \tag{2}$$

$$\mathbf{c}_h = \sum_{t=1}^{T} \mathbf{A}_{ht} \mathbf{V}_t \tag{3}$$

$$\mathbf{c} = \mathrm{concat}(\mathbf{c}_1, \ldots, \mathbf{c}_H) \tag{4}$$

Here, $\mathbf{Q} \in R^{D\times H}$ contains the learnable query vectors for each of the H attention heads, $\mathbf{A} \in R^{T\times H}$ is the attention matrix, and $\mathbf{c}_h \in R^{1\times D}$ and $\mathbf{c} \in R^{1\times HD}$ are the sub-representations and the final utterance-level speaker representation, respectively.

The key idea behind MHFA is that the keys $\mathbf{K}$ capture phonetic information, allowing each attention head to focus on a specific set of phonetic units. Simultaneously, the values $\mathbf{V}$ encode speaker discriminative information, ensuring that the final representation $\mathbf{c}$ is conditioned on both speaker and phonetic characteristics.

To stabilise the fine-tuning process and improve performance, propose two strategies (Peng et al., 2023):

1. Fine-Tuning Regularisation: An $\mathbf{L}_2$ regularization term is added to the overall loss function, encouraging the fine-tuned model's weights to remain close to the initial pre-trained weights:

$$\mathcal{L} = \mathcal{L}_{spk} + \lambda \sum_{j=1}^{|\Theta|} \|\theta_j - \theta_j^p\|_2^2 \tag{5}$$

Here, $\mathcal{L}_{spk}$ is the speaker classification loss, Θ denotes the model parameters, θ_j^p are the corresponding parameters from the initial pre-trained model, and λ is a hyperparameter controlling the strength of the regularisation.

2. Layer-wise Learning Rate Decay (LLRD): Instead of using the same learning rate for all Transformer layers during fine-tuning, LLRD assigns lower learning rates to the bottom layers and higher rates to the top layers, as follows:

$$\mathrm{LR}_l = \mathrm{LR}_1 \cdot \xi^{l-1} \tag{6}$$

Here, LR_l is the learning rate for the l-th Transformer layer, LR_1 is the base learning rate for the bottom layer, and ξ is a weight decay factor controlling the rate of increase in learning rates across layers.

The authors demonstrate that these fine-tuning strategies, combined with the MHFA backend, achieve state-of-the-art performance in speaker verification while significantly reducing training time compared to previous approaches.

3 Proposed Method

The proposed method is inspired by the research presented in Deep Information Maximiser (Hjelm et al., 2019). Here we introduce a regularisation mechanism aimed at increasing the mutual information between the high-level final embedding and the frame-level features. This enhancement seeks to retain valuable information from the frame-level features.

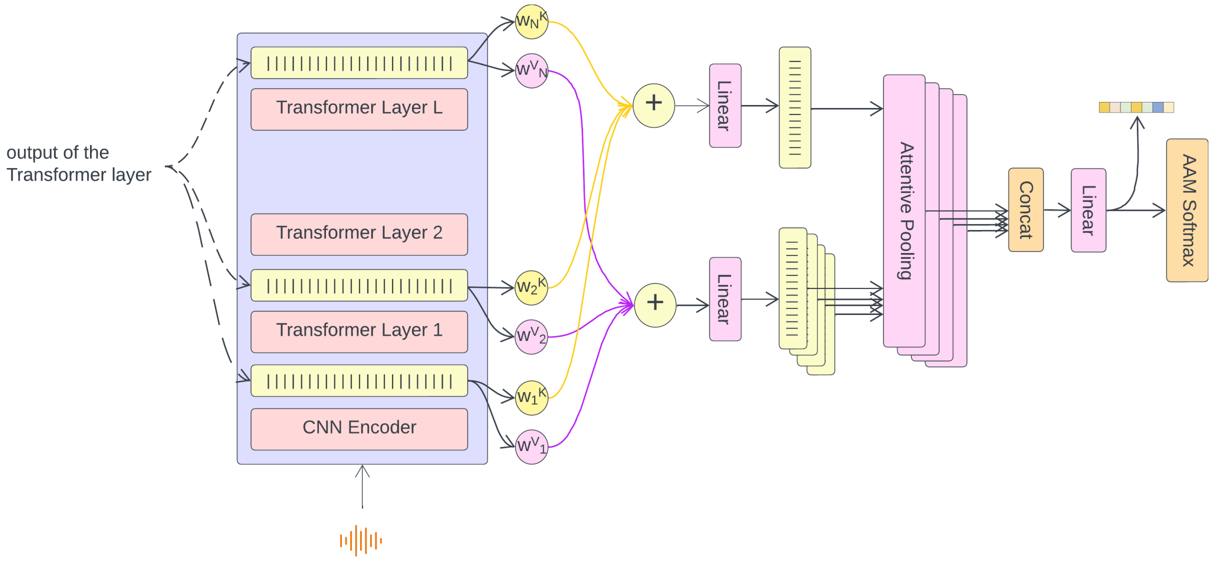


Figure 1: baseline model (Peng et al., 2023)

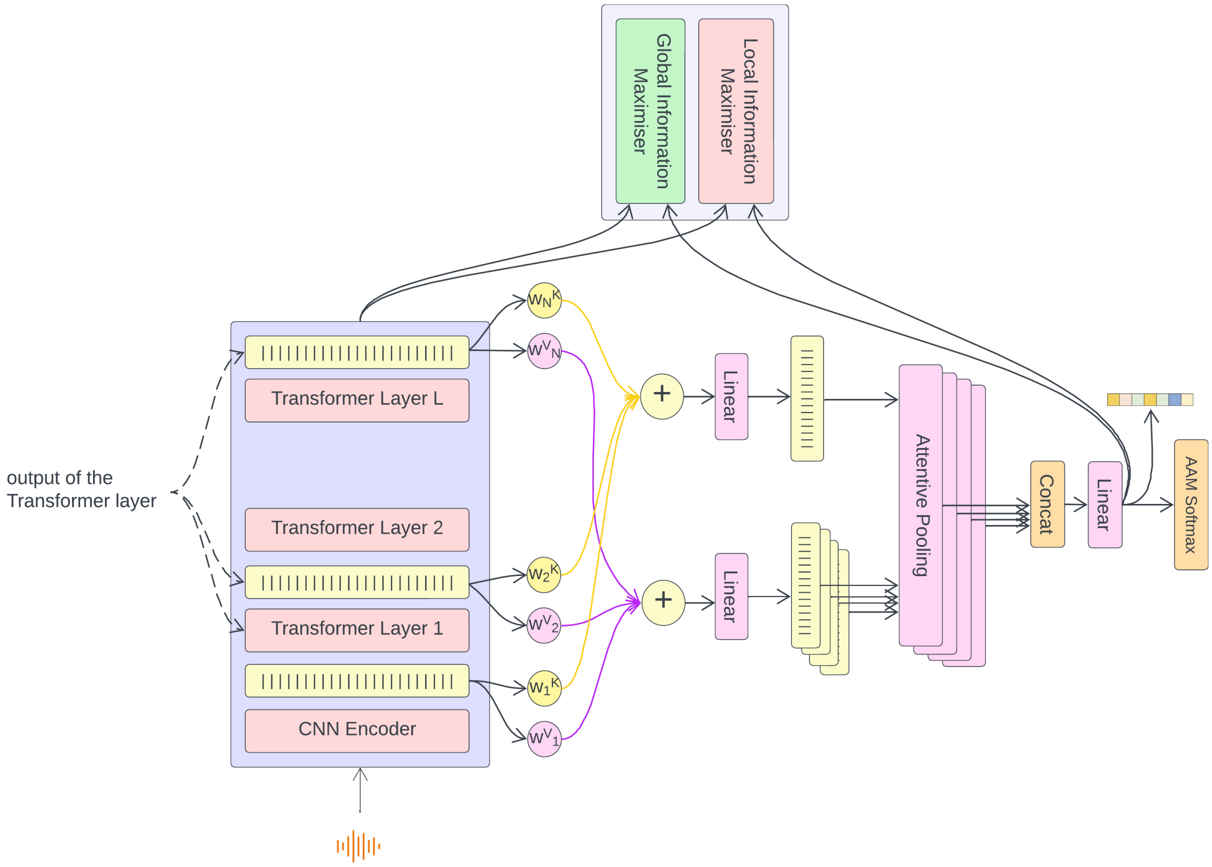


Figure 2: Proposed Model

An additional discriminator, taking a pair of frame-level and high-level embedding as input, is introduced to estimate and maximise the mutual information between these two sets of features. This process effectively functions as a regularising term for the pooling layer, encouraging the embedding vector to capture as much meaningful information from the frame-level features as possible. The discriminator, which functions as a neural network, assesses whether a given concatenated pair of frame-level (low-level) feature maps and high-level embedding corresponds, thereby estimating the common information shared between them.

Two specific discriminators, Global InfoMax (GIM) and Local InfoMax (LIM), which are tailored to capture local and global structures, are employed to estimate and maximise local and global mutual information, respectively.

3.1 Global Information Maximisation (GIM)

Global Information Maximisation (GIM) seeks to increase the mutual information between the output feature map from the ASR encoder and the final speaker embedding. This approach is designed to enhance consistency and contextual understanding within the speaker verification process. GIM optimises $E_\psi : X \to Y$ with neural network parameters ψ, by maximising the mutual information $\mathcal{I}(X, E_\psi(X))$ between X and $E_\psi(X)$. Here, X is the intermediate feature map, and $E_\psi(X)$ is the final embedding created after pooling.

$$(\hat{\omega}, \hat{\psi})_G \in \arg\max_{\omega,\psi} \hat{\mathcal{I}}_\omega(X; E_\psi(X)), \qquad (7)$$

To achieve this, GIM flattens the ASR transformer's feature maps along the feature axis and then concatenates them with the final speaker embedding. Based on this concatenation, GIM assigns a score to measure the mutual information, thereby providing a more accurate representation of the speaker's unique characteristics.

3.2 Local Information Maximiser (LIM)

While GIM can introduce irrelevant dependencies, such as noise, that are not useful for classification, the Local Information Maximiser (LIM) addresses this by focusing on maximising the average mutual information between the high-level embedding and all local frames of the feature map. This approach encourages high-level representation to maintain high mutual information with all frames, promoting the encoding of aspects of data that are shared across frames.

LIM optimises E_ψ with neural network parameters ψ, by maximising the average mutual information $\mathcal{I}(X, E_\psi(X))$ between all the frames F and $E_\psi(X)$. Here, X represents the intermediate feature map, and $E_\psi(X)$ is the final embedding created after pooling.

$$(\hat{\omega}, \hat{\psi})_L = \arg\max_{\omega,\psi} \frac{1}{F} \sum_{i=1}^{F} \mathcal{I}_{\omega,\psi}(x_i; E_\psi(X)) \qquad (8)$$

In this formulation, the final embedding is concatenated with each frame of the ASR transformer feature map (intermediate representation). By maximising mutual information between each local frame and the high-level embedding, LIM ensures that the high-level embedding captures the most relevant and shared information across all frames, enhancing the robustness and accuracy of the classification task.

4 Loss Function

Both LIM and GIM are applied together to train the model, optimising the classification loss during training. The overall loss function can be described as follows:

$$\begin{aligned} L_{Total} &= L_{Classification} \\ &+ \alpha \hat{I}_{\omega_G,\psi}(X; E_\psi(X)) \\ &+ \beta \hat{I}_{\omega_L,\psi}(x_i; E_\psi(X)) \end{aligned} \qquad (9)$$

The first term, $L_{Classification}$, is the speaker classification loss. The Additive Angular Margin (AAM) Softmax loss function is used as the classification loss. The second and third terms are the global and local MINE objectives ω_G and ω_L are the parameters for the global and local discriminators, respectively). These MINE objectives act as regularisation terms with weights α and β during the training of the entire system. The total loss function is calculated as follows:

$$\begin{aligned} &\hat{T}^{(JSD)}_{\omega,\psi}(X; E_\psi(X)) = \\ &E_P[-\text{sp}(-T_{\psi,\omega}(x, E_\psi(x)))] \\ &-E_{\tilde{P}}[\text{sp}(T_{\psi,\omega}(x', E_\psi(x)))] \end{aligned} \qquad (10)$$

The Jensen-Shannon Divergence (JSD) is used as the objective function for MINE. It returns the

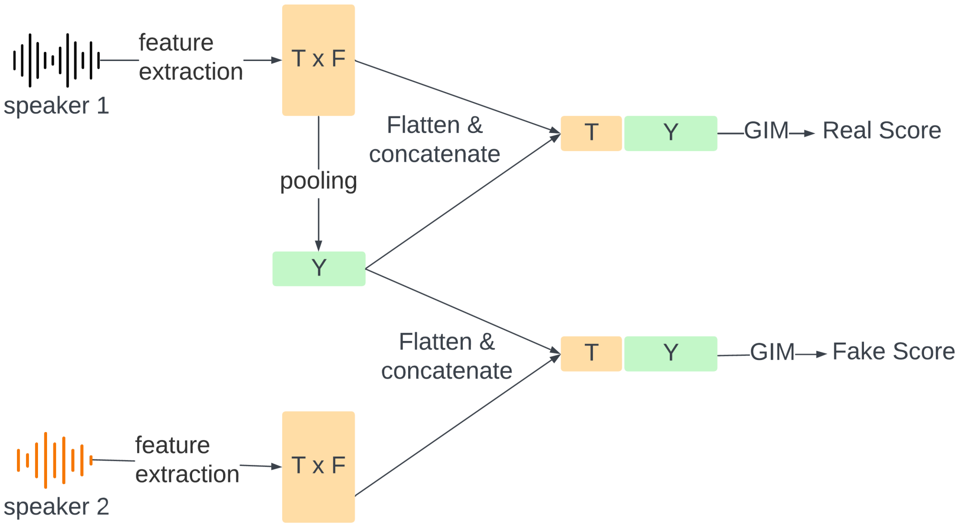


Figure 3: Global Information Maximiser

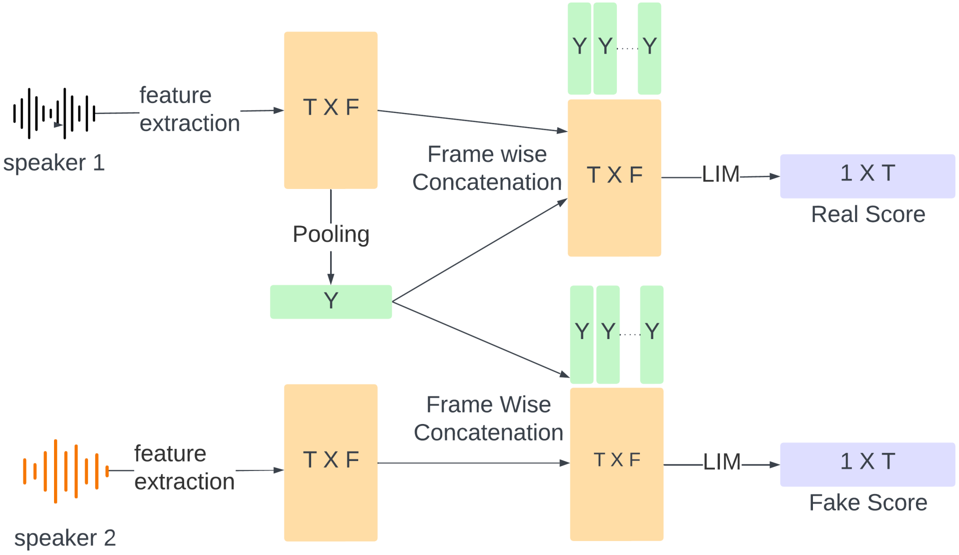


Figure 4: Local Information Maximiser

difference between the softmaxed estimated mutual information of positive pairs (marginal distribution) and the softmaxed estimated mutual information of negative pairs. The JSD provides better and more stable results (Ravanelli and Bengio, 2019) compared to the Kullback-Leibler (KL) divergence used by Belghazi et al.

5 Experiments

5.1 Experiment Setup

To train our proposed model, we utilised the VoxCeleb1 development set (Nagrani et al., 2017), a widely recognised large-scale dataset for text-independent speaker verification. We evaluated the model's performance using the VoxCeleb1 test corpus. For the ASR transformer, we employed the WaveLM-Base-Plus (Chen et al., 2022) model due to its strong performance in previous studies (Peng et al., 2023). The transformer's output had dimensions of 150 x 768, with 150 representing the total number of frames and 768 representing the feature dimension for each frame. Model training was conducted on two 16GB NVIDIA Tesla GPUs in a distributed manner, with a batch size of 32. We conducted experiments both with and without the Deep Information Maximisation (DIM) method, adjusting parameters such as α, β.

The learning rate was set to 0.0001, with a decay rate of 0.95. Both the Local Information Maximiser (LIM) and the Global Information Maximiser (GIM) were implemented using one-

Layer	in channels	out channels	feature dimension	kernel size
conv 1	768	256	150	3
conv 2	256	64	148	3
fc 1	9600	512	-	-
fc 2	512	1	-	-

Table 1: Layer configuration for GIM

Layer	in channels	out channels	feature dimension	kernel size
conv 1	256 + 768	512	150	1
conv 2	512	512	150	1

Table 2: Layer configuration for LIM

dimensional Convolutional Neural Networks (1D CNNs) and fully connected layers.

We used one-dimensional Convolutional Neural Networks (1D CNNs) for both Local and Global Information Maximisation (InfoMax) because; firstly, audio data is inherently sequential, with each time step represented by a feature vector, making 1D convolutions ideal for capturing temporal dependencies and local patterns along the time axis. This approach also reduces computational complexity and the number of model parameters compared to 2D convolutions, enhancing efficiency.

5.2 Experiment Results

The experiment evaluated the proposed Deep Information Maximisation (DIM) approach integrated with an attention-based backend for text-independent speaker verification. The primary metric used for performance evaluation was the Equal Error Rate (EER), where a lower EER indicates better performance.

No attention heads	EER
1	1.877
2	1.681
4	1.612
8	1.485
16	1.419
32	1.336

Table 3: Experimental Results for baseline model with different number of attention heads

As the number of attention heads increases, the EER consistently decreases, demonstrating that using more attention heads improves the accuracy of the speaker verification system. The lowest EER of 1.336 is achieved with 32 attention heads.

No of attention heads	Baseline EER	DIM integrated baseline EER
1	1.877	1.845
2	1.681	1.677
4	1.612	1.538
8	1.485	1.441
16	1.419	1.389
32	1.336	1.392

Table 4: Comparison of Baseline and DIM integrated baseline for different attention heads

	$\beta = 0.01$	$\beta = 0.05$	$\beta = 0.1$
$\alpha = 0.01$	1.8664	1.5376	1.7656
$\alpha = 0.05$	1.6278	1.7073	2.0308
$\alpha = 0.1$	1.7709	1.7232	1.9618

Table 5: Experiment results(EER) of the DIM integrated base for different α and β values with four attention heads.

The DIM method was tested with various configurations of the hyperparameters α and β, which control the weights of the global and local mutual information maximisation terms, respectively. The results are presented in the table below, comparing different values of α and β with the baseline model, which had an EER of 1.612.

The experiment results indicate that the integration of the DIM method can improve the performance of the speaker verification system. The configuration with α=0.01 and β=0.05 achieved the best EER of 1.5376, which is an improvement over the baseline EER of 1.612.

In high levels, increasing the number of attention heads generally leads to lower EER, indicating better performance. With up to 16 attention heads, DIM-integrated models outperform the baseline.

Optimal values for α and β significantly impact performance, with lower values generally resulting in better EER, with the optimal combination being α=0.01 and β=0.05. Compared to α, the β has a stronger influence on EER, which emphasizes the importance of Local InfoMax. Overall, the DIM integrated baseline shows consistent improvements over the baseline, confirming the effectiveness of the DIM integration.

Possible Reasons for Variations in Experimental Results for baseline and DIM Integrated in 32 attention heads. (1) Increasing the number of attention heads enhances the model's ability to capture detailed speaker-specific characteristics by simultaneously focusing on multiple aspects of the input features. While this can reduce the Equal Error Rate (EER), it also increases model complexity, which can lead to overfitting if the training data lacks sufficient diversity. (2) There appears to be a saturation point beyond which adding more attention heads does not significantly improve performance. Beyond this point, additional attention heads provide diminishing returns, potentially leading to inefficiency and a low performance/cost ratio. (3) The DIM method aims to maximise mutual information between frame-level and high-level features, preserving discriminative features essential for effective speaker verification. In models with 32 attention heads, the high-dimensional space created can make it difficult for DIM to preserve and maximise relevant information without introducing noise or redundancy, thus conflicting with the model's complexity. (4) The hyperparameters α and β are crucial for model performance as they control the emphasis on mutual information maximisation. Lower values are generally preferred to prevent the model from overly focusing on mutual information at the expense of classification accuracy. However, if set too low, the model might not fully leverage the benefits of Deep InfoMax.

6 Conclusion

The research investigated the use of Deep Information Maximisation (DIM) to mitigate information loss in text-independent speaker verification systems, focusing on the impact of attention heads and DIM integration on performance. Findings revealed that Local Information Maximisation (LIM) plays a significantly larger role than Global Information Maximisation (GIM) in maximising mutual information, highlighting the importance of preserving local context for accuracy improvement.

Experiments showed that up to 16 attention heads, the DIM-integrated model outperformed the baseline by reducing the Equal Error Rate (EER). Beyond this point, the EER increased, indicating limitations in handling higher complexity and potential noise introduction. This suggests that while DIM is beneficial, its integration with numerous attention heads requires careful balancing to avoid overfitting and diminishing returns. The results emphasize the potential of attention mechanisms in capturing detailed speaker-specific characteristics but also underline the need to manage model complexity for optimal performance.

7 Future Work

Future research should aim to identify the optimal number of attention heads to balance model complexity and performance, involving further experiments and validation across diverse datasets, introducing data augmentation techniques, such as noise addition, which will allow us to evaluate the robustness and generalisation capability of our proposed method in more challenging and realistic conditions.

Advanced techniques to integrate DIM with more attention heads should be explored, including refining mutual information maximisation and incorporating additional regularisation to reduce noise.

Robust hyperparameter tuning for α and β is crucial. Studies should explore a broader range of these parameters to better understand their impact and identify the most effective configurations. Finally, future work should address the computational demands of training models with many attention heads by optimising training stability and efficiency or exploring alternative architectures.

References

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022.

Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification.

Hongcan Gao, Xiaolei Hou, and Jing Xu. 2020. Vector-Based Attentive Pooling for Text-Independent Speaker Verification. In *Interspeech 2020*, pages 936–940. ISCA.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

Seong-Hu Kim, Hyeonuk Nam, and Yong-Hwa Park. 2022. Decomposed Temporal Dynamic CNN: Efficient Time-Adaptive Network for Text-Independent Speaker Verification Explained with Speaker Activation Map. ArXiv:2203.15277 [cs, eess].

Tianchi Liu, Rohan Kumar Das, Kong Aik Lee, and Haizhou Li. 2022. Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7517–7521.

Aryan Mobiny and Mohammad Najarian. 2018. Text-Independent Speaker Verification Using Long Short-Term Memory Networks. ArXiv:1805.00604 [cs, eess].

A. Nagrani, J. S. Chung, and A. Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*.

Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. 2018. Attentive Statistics Pooling for Deep Speaker Embedding. In *Interspeech 2018*, pages 2252–2256. ArXiv:1803.10963 [cs, eess].

Junyi Peng, Oldřich Plchot, Themos Stafylakis, Ladislav Mošner, Lukáš Burget, and Jan Černocký. 2023. An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 555–562.

Mirco Ravanelli and Yoshua Bengio. 2019. Learning speaker representations with mutual information. pages 1153–1157.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, Calgary, AB. IEEE.

Yun Tang, Guohong Ding, Jing Huang, Xiaodong He, and Bowen Zhou. 2019. Deep speaker embedding learning with multi-level pooling for text-independent speaker verification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6116–6120.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, Florence, Italy. IEEE.

Yong Zhao, Tianyan Zhou, Zhuo Chen, and Jian Wu. 2020. Improving Deep CNN Networks with Long Temporal Context for Text-Independent Speaker Verification. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6834–6838. ISSN: 2379-190X.

Tianyan Zhou, Yong Zhao, Jinyu Li, Yifan Gong, and Jian Wu. 2019. CNN with Phonetic Attention for Text-Independent Speaker Verification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 718–725, SG, Singapore. IEEE.

Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey. 2018. Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification. In *Interspeech 2018*, pages 3573–3577. ISCA.

Context-Aware Question Answering in Urdu

Samreen Kazi and **Shakeel Khoja**
School of Mathematics and Computer Science
Institute of Business Administration (IBA)
Karachi, Pakistan
{sakazi, skhoja}@iba.edu.pk

Abstract

Answer sentence identification and extractive answer span identification are crucial components in the development of robust question-answering (QA) systems. Despite advancements in natural language processing (NLP), there remains a gap in applying these technologies to Urdu due to the scarcity of annotated datasets and linguistic tools. This paper addresses this gap by introducing a three-stage unified framework aimed at improving both tasks.The framework consists of three key components: key sentence identification, extractive answer span identification, and a unified scoring model. For sentence identification, the framework employs a sliding window approach for text alignment, using noun term frequency for relevance scoring and vector similarity from pre-trained word embeddings to capture deeper semantics. For extractive answer span identification, the model uses a fine-tuned multilingual BERT (mBERT) model trained on the Universal Dependencies (UD) Treebank for Urdu to identify noun chunks for linguistic relevance. The unified model integrates probabilities from sentence identification and span extraction to derive a composite score for selecting the most relevant answer span.Experimental results show the proposed approach significantly outperforms traditional methods, demonstrating its potential for broader application in other low-resource languages like Urdu.

1 Introduction

Question Answering (QA) systems are essential tools for extracting precise information from large text corpora in response to user queries (Kazi et al., 2023). Developing such systems for low-resource languages like Urdu is particularly challenging due to the lack of extensive annotated datasets and specialized linguistic tools (Daud et al., 2017). Standard QA approaches, which often rely on syntactic and semantic similarities typical of high-resource languages, struggle to capture the linguistic nuances and rich morphology characteristic of Urdu. This gap highlights the need for methodologies tailored specifically to the unique challenges posed by such languages (Otegi et al., 2020). Answer sentence identification and answer extraction are critical components of QA systems. Answer sentence identification involves identifying sentences based on their likelihood of containing the correct answer, while answer extraction focuses on identifying the specific text segment within these sentences that directly answers the question (Allam and Haggag, 2012). Traditional models for these tasks often fall short as they rely predominantly on surface-level syntactic and semantic similarities, which are insufficient for capturing the complex linguistic features of Urdu (Chang et al., 2024). This paper introduces a comprehensive three-stage unified framework that integrates Key sentence identification, extractive answer extraction to enhance performance for Urdu text. The proposed model employs combination of traditional and advanced text processing techniques to address the challenges posed by the Urdu language. The first stage utilizes a custom-designed weighted sliding window algorithm (Richardson et al., 2013) for precise text alignment, enhancing relevance scoring through the term frequency of nouns. The second stage leverages a fine-tuned multilingual BERT (mBERT) model (Devlin et al., 2018), trained on the Universal Dependencies (UD) Treebank for Urdu (Bhat et al., 2017), to identify noun chunks within the text. These chunks are evaluated and grouped based on semantic similarity, with the best chunk being selected based on aggregated scores. The final stage combines the probabilities from both the identification and extraction stages into a unified score, ensuring the identification of the most relevant answer chunk from the top-ranked sentences by leveraging both sentence-level and phrase-level evidence. The research contributions of this work are as follows:

1. Development of a three-stage unified framework that integrates key sentence identification and extractive answer span identification, specifically tailored for the Urdu language.
2. Introduction of a customized sliding window algorithm for question-passage alignment, enhancing relevance scoring through the term frequency of nouns.
3. Demonstration of significant performance improvements over traditional methods on Urdu datasets, highlighting the model's potential for broader application in other low-resource languages.

The rest of this paper is structured as follows. Section 2 provides an overview of the relevant background. Section 3 details our methodology, focusing on the stages of answer sentence identification, answer extraction, and the unified model for QA. Section 4 outlines the experimental setup, and Section 5 presents the results, followed by a discussion of their significance.

2 Literature Review

Question Answering (QA) systems have advanced significantly in high-resource languages like English, Chinese, and European languages. Early multi-stage QA methods relied on feature engineering and traditional machine learning. Yao et al. (Yao et al., 2013) used syntactic features and logistic regression for answer ranking, highlighting linguistic structure's role.With deep learning, neural models became prominent. Severyn and Moschitti (Severyn and Moschitti, 2015) introduced a CNN for sentence pair modeling, outperforming previous methods. The advent of transformer models, notably BERT (Devlin et al., 2018), revolutionized QA. Nogueira and Cho (Nogueira and Cho, 2019) fine-tuned BERT for passage ranking, setting new benchmarks.Answer extraction has evolved from rule-based systems like TextRunner (Banko et al., 2007) to neural models. Named Entity Recognition (NER) significantly aids this process, with Lample et al. (Lample et al., 2016) combining LSTMs and CRFs. Span-based extraction models, like SpanBERT (Joshi et al., 2020), further improved extractive QA tasks.End-to-end QA systems like DrQA (Chen et al., 2017) have shown strong performance, supported by datasets like SQuAD (Rajpurkar et al., 2016), which have become standard benchmarks. The introduction of datasets such as Natural Questions (Kwiatkowski et al., 2019) has further pushed open-domain QA research.These advancements have inspired research in low-resource languages like Urdu, Arabic, and Hindi (Kazi and Khoja, 2021) (Arif et al., 2024) (Shaheen and Ezzeldin, 2014) (Gupta et al., 2018). While transformer models like T5 (Raffel et al., 2020) have been adapted, challenges remain in effectively applying these to Urdu due to linguistic nuances and resource constraints. Our work introduces a lightweight, interpretable multi-stage framework leveraging traditional techniques alongside fine-tuned multilingual BERT, addressing Urdu-specific challenges. Although it may not match the accuracy of models like mT5, it provides a foundation for advanced hybrid systems.

3 Methodology

This section presents the two-stage approach used to integrate key sentence identification and extractive answer span identification into a unified learning model, as illustrated in Figure 1. The methodology employs a sliding window technique for measuring text overlap between the passage and the question, incorporates term frequency of nouns for relevance scoring, and computes semantic vector similarity using word embeddings. Additionally, a fine-tuned mBERT model, trained on the UD Treebank for Urdu, is used for high-quality chunk identification. The framework consists of the following three components:

(i) Key sentence identification: A probabilistic model is used to identify sentences in

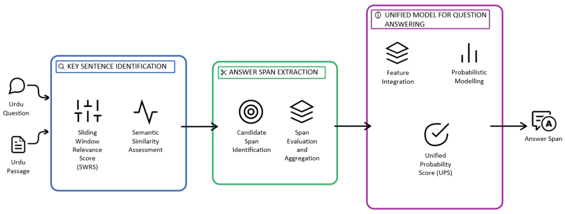

KEY SENTENCE IDENTIFICATION
Urdu Question
Urdu Passage
Sliding Window Relevance Score (SWRS)
Semantic Similarity Assessment
ANSWER SPAN EXTRACTION
Candidate Span Identification
Span Evaluation and Aggregation
UNIFIED MODEL FOR QUESTION ANSWERING
Feature Integration
Probabilistic Modelling
Unified Probability Score (UPS)
Answer Span

Figure 1: Overall Architecure of Context-Aware QA for Urdu Language

the passage that are most relevant to the question.

(ii) Extractive Answer Span Identification: Another probabilistic model is used to extract answer spans from the identified key sentences.

(iii) Unified model: The outputs from the key sentence identification and answer extraction stages are integrated into a unified model. The probabilities from both task-specific models are combined to improve the overall performance of the system.

3.1 Key Sentence Identification

In this section, we describe the methods used for key sentence identification, which involves determining which sentences are most likely to contain the correct answer to a given question. This process is divided into two main parts: Sliding Window Relevance Score Feature(SWRS) and Semantic Similarity Feature. The SWRS feature identifies the most relevant segment of the passage using a sliding window approach, while the semantic vector similarity measures the similarity between the question and candidate sentences using word embeddings.After extracting both features logistic regression model predicts the probability that each sentence contains an answer.

3.1.1 Sliding Window Relevance Score Feature(SWRS)

The SWRS feature begin by tokenizing the question and passage into individual words using UrduHack (ALi, 2020). This allows for a detailed comparison at the word level. Next, the term frequency (TF) for nouns in the corpus is calculated, as nouns often carry significant meaning in sentences. Using a sliding window approach, the passage is segmented into overlapping windows of a fixed size, and the relevance score for each window is calculated based on the overlap with the question words and the term frequency of nouns within the window. Additionally, a word co-occurrence matrix is used to enhance the relevance scoring by considering the contextual relationships between words. The window with the highest relevance score is then selected as the most aligned segment, providing a focused area of the passage that is most likely to contain the answer , as described in Algorithm 1.

Algorithm 1 Algorithm of SWRS

```
Abbreviations:
    • $Q_{\text{text}}$: Question string
    • $P_{\text{text}}$: Passage string
    • $W$: Window size
    • $s$: Step count
    • $C\_m$: Co-occurrence matrix
    • $TF_{\text{noun}}$: Term Frequency of nouns
    • $I_{\text{C}}(n)$: Inverse Count of nouns
Input:
    • $Q_{\text{text}}$: Question string
    • $P_{\text{text}}$: Passage string
    • $W$: Window size
    • $s$: Step count
Tokenization:
    • Split $Q_{\text{text}}$ and $P_{\text{text}}$ into words
    • Output: $Q_{\text{tokens}}$, $P_{\text{tokens}}$
Calculate TF for Nouns:
    • Identify nouns in $P_{\text{tokens}}$ and calculate $TF_{\text{noun}}$
Calculate Co-occurrence Matrix:
    • Compute $C\_m$ using $P_{\text{tokens}}$ and $Q_{\text{tokens}}$
Overlap Score ($O_{\text{s}}$):
    • For each window $W$ in $P_{\text{tokens}}$:
        – $O_{\text{s}} = \sum_{n \in W \cap Q_{\text{tokens}}} TF_{\text{noun}}(n) \times I_{\text{C}}(n)$
Co-occurrence Score ($Co_{\text{s}}$):
    • For each window $W$ in $P_{\text{tokens}}$:
        – $Co_{\text{s}} = \sum_{w \in W} \sum_{q \in Q_{\text{tokens}}} C\_m(w, q)$
Relevance Score ($R_{\text{s}}$):
    • For each window $W$ in $P_{\text{tokens}}$:
        – $R_{\text{s}} = O_{\text{s}} + Co_{\text{s}}$
Sliding Window:
    • Slide $W$ across $P_{\text{tokens}}$ with size $w$ and step $s$.
    • Calculate scores and find $j^* = \arg\max_j R_{\text{s}}(j)$
Output:
    • $best\_window = W_{j^*}$
Return:
    • $best\_window$
```

3.1.2 Semantic Similarity Features

The initial phase involves utilizing pre-trained word embeddings, specifically fastText embeddings (Bojanowski et al., 2016) trained on a large corpus of question-answer pairs. FastText embeddings are preferred here as they incorporate subword information, capturing morphological nuances and effectively handling out-of-vocabulary words. Training on a QA-specific corpus ensures that the embeddings are tailored to the domain, enhancing their representation of relevant semantic relationships. Word embeddings map words into a continuous vector space, where semantically similar words are situated closer together. Each word w in the question and candidate sentences is transformed into a high-dimensional vector $\mathbf{v}_w$ that encapsulates its semantic nuances. This transformation captures the contextual meaning of words, facilitating a sophisticated comparison between the question and candidate sentences beyond mere lexical similarity.Subsequently, a single vector representation for the entire question and each candidate sentence is constructed by aggregating the vectors of content words, such as nouns, verbs, and adjectives. This aggregation, achieved through vector summation:

$$\mathbf{V}_{\text{sentence}} = \sum_{w \in \text{content words}} \mathbf{v}_w$$

integrates the semantic information of all content words, resulting in a composite vector that represents the overall meaning of the sentence. This method enhances the capacity to perform meaningful comparisons between the question and the candidate sentences. The final step involves computing semantic similarity by calculating the cosine similarity between the vector representation of the question $\mathbf{V}_Q$ and each candidate sentence $\mathbf{V}_C$:

$$\text{Cosine Similarity} = \frac{\mathbf{V}_Q \cdot \mathbf{V}_C}{\|\mathbf{V}_Q\| \|\mathbf{V}_C\|}$$

3.2 Extractive Answer Span Identification

In this section, we focus on the process of extracting the specific span of the sentence that answers the given question. The extraction process ensures that the most relevant and precise text span is identified and selected, providing an accurate response to the user's query.In the end, another logistic regression model evaluates the likelihood of each candidate span being the correct answer.

3.2.1 Candidate span extraction

In the answer extraction process, the initial step involves identifying candidate spans within the sentences that are likely to contain the correct answer. To achieve this, we utilize a fine-tuned mBERT (multilingual BERT) model, specifically trained on the Universal Dependencies (UD) Treebank for Urdu. This model is proficient in identifying high-quality noun phrases and other relevant text segments, ensuring that the candidate span are both linguistically coherent and contextually appropriate for further evaluation. Let C represent the set of candidate span identified as:

$$C = \{c_1, c_2, \ldots, c_n\}$$

where each c_i represents an individual candidate span. Once the candidate spans are identified, the next step is to evaluate the quality of each span based on several features. These features include the length of the span, its position within the sentence, and its relevance to the question posed. The evaluation process can be represented by scoring each span c_i as follows:

$$S(c_i) = \alpha \cdot \text{len}(c_i) + \beta \cdot \text{pos}(c_i) + \gamma \cdot \text{rel}(c_i, Q)$$

where α, β, γ are weighting factors that adjust the importance of each feature, and Q is the vector representation of the question. This scoring helps determine the likelihood that a given span contains the correct answer, allowing us to filter and retain only the most promising candidates for further consideration. After evaluating the individual span, the subsequent step is to group semantically equivalent span. This grouping is based on both word overlap and semantic similarity, ensuring that span conveying the same or similar information are clustered together. Let G represent these groups:

$$G = \{g_1, g_2, \ldots, g_m\}$$

where each group g_j contains semantically similar chunks. Semantic similarity between span c_i and c_j can be computed using:

$$\text{sim}(c_i, c_j) = \frac{\mathbf{v}_{c_i} \cdot \mathbf{v}_{c_j}}{\|\mathbf{v}_{c_i}\| \|\mathbf{v}_{c_j}\|}$$

where $\mathbf{v}_{c_i}$ and $\mathbf{v}_{c_j}$ are the vector representations of the spans. This step ensures that we consolidate the information across similar span, which aids in aggregating their scores.The final step in the span extraction process involves selecting the best span from each group of equivalent span. This is achieved by aggregating the scores within each group and selecting the span with the highest score:

$$c^* = \arg\max_{c_i \in g_j} S(c_i)$$

This selection process ensures that the chosen span not only aligns well with the question but also represents the most reliable and precise part of the text. By considering aggregated scores, we enhance the robustness of our selection, ensuring the extracted answer is both relevant and accurate. This systematic approach, encompassing candidate span identification, span evaluation, grouping of equivalent chunks, and the final selection of the best span, guarantees that the extracted span is contextually appropriate and precise. This enhances the overall effectiveness of the question-answering system by ensuring that semantically similar sentences, even without shared lexical items with the question, are considered relevant, thereby significantly improving the accuracy of the answer retrieval.

3.3 Unified Model for Question Answering

In this section, we introduce the methodology for combining probabilities derived from the key sentence identification and span extraction processes. The objective is to unify these probabilities into a single score that can identify the most relevant answer span from the top-ranked sentences. By integrating both identification and extraction stages, we ensure that the selected span is contextually appropriate and precisely extracted. Our unified model leverages advanced feature engineering and probabilistic modeling to enhance the accuracy and relevance of extracted answers from textual data. This sophisticated approach combines the strengths of key sentence identification and extractive answer span identification, tailored specifically to the nuances of different question types in Urdu. Here's an overview of the methodology:

3.3.1 Feature Extraction

The model leverages features extracted from various modules, including those specifically designed for key sentence identification and extractive answer span identification. Additionally, it incorporates a diverse set of features tailored to capture both the lexical and semantic nuances of the text, further enhancing its ability to identify the most relevant answer:

Question Type Specific Features:

- **Question-word Features**: Extracts and utilizes the POS, DEP, and NER tags of the main question word (e.g., کیا, کون/کس), appending these to the question type to refine feature sensitivity.
- **Question Focus**: Determines the focus noun phrase within the question, crucial for aligning the model's attention to the most relevant part of the quer.

Query Ques:

Pairs the headword and question focus features, creating compound indicators such as question-type|question-focus-word|headword-pos-tag.

Span Tags:

checking for the presence of significant noun phrase that match expected answers based on the question type.

3.3.2 Conditional Probability Learning

During training, the model learns the conditional probabilities $P(c \mid s, f)$:

- c: Candidate answer span.
- s: Sentence containing the answer span.
- f: Feature vector encompassing all extracted features for the answer span and the sentence.

This probabilistic framework captures the complex interdependencies between sentence relevance, answer span quality, and the rich contextual features derived from the text.

3.3.3 Unified Probability Score

In the prediction stage, the model calculates a unified probability score(UPS) for each candidate answer span:

$$\text{UPS}(c) = P(c \mid f) \times P(s \mid c, f)$$

where $P(c \mid f)$ represents the probability of the answer span being correct, based solely on its features, and $P(s \mid c, f)$ assesses the conditional probability that the sentence is key, given the span and its features.

3.3.4 Final Answer Selection

The model selects the candidate answer span with the highest unified probability score. This selection process prioritizes spans that are not only plausible based on their intrinsic features but also originate from sentences that are contextually aligned with the question. This dual consideration ensures that the chosen answers are both accurate and contextually relevant, thereby significantly enhancing the performance of the question-answering system.

4 Experiments

4.1 Data

For the experimental validation of our unified model, we employed two significant datasets tailored for Urdu question-answering systems: UQuAD and UQA. These datasets are selected and adapted to rigorously test both the answer sentence identification and extraction capabilities of our model. The Urdu Question Answering Dataset (UQuAD1.0) includes 46,481 Stanford Question Answering Dataset(SQuAD 2.0) (Rajpurkar et al., 2016) questions translated using google translation API covering various domains such as history, science, and general knowledge. It also contains 4000 crowdsource question annotated by humans based on Question types. The UQA corpus on the other hand features 136,211 questions, focusing on domain-specific topics, created using the "Enclose to Anchor, Translate, Seek" (EATS) technique from the Stanford Question Answering Dataset (SQuAD 2.0). This technique ensures that answer spans are preserved in the translated context paragraphs, making it suitable for training and evaluating Urdu QA models. It consists of 83,018 answerable and 41,727 unanswerable questions, providing a balanced setup for models to not only retrieve accurate answers but also to discern when no plausible answer is present in the text.Table**??**shows distrbution of dataset.

Dataset	QA Pairs	Question Types	EM
UQuAD (MT)	45,000	No	0.66
UQuAD (CS)	4,000	Yes	0.50
UQA (MT)	124,745	No	0.85

Table 1: Distribution of UQA and UQuAD Datasets.
'MT' = Machine Translation, 'CS' = Crowd-Sourced.

4.1.1 Dataset Adaptation for Sentence identification Evaluation:

To assess the identification capabilities of our model adequately, we adapted both UQuAD and UQA by an approach that includes:

1. Extraction of Candidate Answer Sentences: We analyzed each paragraph within the datasets to identify all sentences that could potentially contain the answer, based on their content overlap with the gold-standard answer provided (Charras and Lecroq, 2004).

2. Annotation of Candidate Sentences: Each identified sentence was subsequently labeled as either '1' (containing the answer) or '0' (not containing the answer). This binary annotation serves as the definitive ground truth for the key sentence identification task.

The adapted dataset was divided into training, and test sets as shown in Table 2, ensuring that no question-paragraph pair appeared in multiple subsets.

Dataset	Train	Test
UQuAD(MT)	36,000	9,000
UQA	99,796	24,949

Table 2: Train/Test Split for Training Model

4.1.2 Evaluation Metrics

We adhere to standard evaluation procedures and metrics for QA rankers as outlined in prior research (Rajpurkar et al., 2016). Our evaluation metrics for assessing the performance of question answering systems include:

- **Exact Match (EM)**: Measures the percentage of predictions that exactly match any one of the ground truth answers.

- **F1 Score**: Computes the harmonic mean of precision and recall at the individual token level, considering both the partial correctness of the answers.
- **Average Precision at K** : Defined as the average of correct answer sentences within the top K results to evaluate Key sentence identification.

4.2 Baseline Models for Comparison

4.2.1 Word N-grams - Sliding Window Baseline

To establish a comparative baseline, we used the Word N-gram overlap method, a traditional technique used to determine textual similarity (Richardson et al., 2013). This method involves segmenting texts into fixed-length N-grams and calculating similarity scores based on the overlap of these N-grams. This approach has been validated in various applications such as plagiarism detection and text reuse (Daud et al., 2017). For our purposes, we adapt it to extract answer spans by tokenizing the text into N-grams and selecting spans based on their overlap with the query, calculated as follows:

$$\text{overlap} = \frac{|S(P_1, n) \cap S(P_2, n)|}{\min(|S(P_1, n)|, |S(P_2, n)|)} \quad (1)$$

4.2.2 TF-IDF - Feature-Based Baseline

Additionally, we employ the traditional TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique, which represents text using term frequency-inverse document frequency metrics. This method is enhanced with N-gram frequencies ranging from unigrams to trigrams to capture local word order, crucial for understanding contextual relevance. The TF-IDF value for a term t in a document d within a document set D is calculated as follows:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \quad (2)$$

where:

- $\text{TF}(t, d)$ is the term frequency of term t in document d, and
- $\text{IDF}(t, D)$ is the inverse document frequency of term t across the document set D, defined as:

$$\text{IDF}(t, D) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right) \quad (3)$$

For N-grams, the terms t are extended to include unigrams, bigrams, and trigrams, thereby enhancing the textual representation by capturing contiguous sequences of up to three words. This enhancement allows for a more nuanced understanding of the text's structure and semantics.

5 Results and Discussion

We employed various baseline approaches as mentioned in section 4.2, which include traditional text representation techniques to retrieve answer.In evaluating the sentence identification capabilities of our model, we observed differentiated performance across various question types, as shown in Table 3. The model exhibited high precision in identifying sentences relevant to 'Who', 'When', and 'Where' questions, achieving AP@K scores of 0.55, 0.58, and 0.68, respectively. These question types typically involve extracting specific entities or temporal and locational information, which are well-captured by our model's feature set. Conversely, 'What' and 'Why' questions, which often require understanding broader contexts or causal relationships, posed greater challenges, reflected in lower AP@K scores of 0.35 and 0.40. 'How' questions, diverse in their structure and intent, showed moderate performance with an AP@K score of 0.44. Overall, the model achieved an average precision across all question types of 0.44, indicating a robust capability to identify relevant answer-containing sentences but also highlighting potential areas for enhancement in handling complex question contexts and reason behind lower accuracy of unified model shown in tables 5 and 4. overall our unified model achieved better results in answer extraction as shown in 5 and 4 showcasing the effectiveness of our unified model compared to the traditional approaches.

5.1 Discussion

The results indicate that our unified Model significantly outperforms the baseline models,

Question Type	Average Precision at K
What (کیا)	0.35
Who (کون)	0.55
When (کب)	0.58
Where (کہاں)	0.68
Why (کیوں)	0.40
How (کیسے)	0.40

Table 3: Performance of the sentence identification model across various question types using Average Precision at K metric.

Dataset	N-gram	TF-IDF	Unified Model
UQuAD	0.15	0.25	0.48
UQA	0.20	0.22	0.55

Table 4: F1: Performance comparison of different models on UQuAD and UQA datasets.

Dataset	N-gram	TF-IDF	Unified Model
UQuAD	0.12	0.28	0.60
UQA	0.10	0.31	0.50

Table 5: EM: Performance comparison of different models on UQuAD and UQA datasets.

demonstrating its efficacy in leveraging complex feature interdependencies to accurately identify and extract answers. This superior performance underscores the advantage of integrating sentence identification with extraction capabilities in a unified model, particularly in the nuanced context of Urdu language question answering. Our approach to integrating sentence identification and span extraction through unified probabilistic modeling has demonstrated promising results. For example, in the UQuAD and UQA datasets, we observed marked improvements in precision over traditional models, as evidenced by the scores illustrated in our performance tables. To better understand the nuances of the model's performance, let's consider practical examples using Urdu question-answer pairs. Imagine a question in Urdu like "محمد علی نے ہونڈا میں کتنے سال کام کیا؟" (How many years did Mohammad Ali work at Honda?). Our model might identify a sentence such as "محمد علی نے ہونڈا میں چالیس سال تک کام کیا۔" (Mohammad Ali worked at Honda for 40 years), scoring it highly due to the direct match of numeric and contextual information. Conversely, sentences without direct numerical answers or only peripheral relevance to Honda and Mohammad Ali would receive significantly lower scores. This method effectively discerns the relevance and specificity of candidate answer sentences. However, when evaluating our system against state-of-the-art transformer-based models, such as those employing BERT or its derivatives, we notice a gap in achieving top-tier performance metrics like Exact Match (EM) and F1. This discrepancy can largely be attributed to the inherent limitations of N-gram and TF-IDF models in capturing the deep semantic structures that transformer models excel at.

Limitations

This study offers valuable insights into applying NLP techniques for Urdu language processing, but it does face limitations. The primary datasets used, UQuAD and UQA, while comprehensive, do not entirely capture the full diversity of Urdu language use due to synthetic nature of data. Additionally, this model focus mainly on syntactic and semantic features and do not extensively address other linguistic elements such as pragmatics and discourse context, which are vital for fully understanding complex questions. Furthermore, despite showing promising results in Urdu, the model's effectiveness in other low-resource or morphologically rich languages have not been explored. This may limit its broader applicability and scalability, especially in contexts where transformer-based models have shown superior performance.

Ethics Statement

This research adheres to the highest ethical standards. All datasets, including text and question-answer pairs, were sourced from publicly accessible repositories. We ensured that no private or sensitive data was utilized without explicit consent. All sources have been meticulously cited, and the use of any copyrighted material complies strictly with applicable legal standards, ensuring transparency and integrity in our research methodology.

References

Ikram ALi. 2020. Urduhack: A python library for urdu language processing.

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. 2012. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).

Samee Arif, Sualeha Farid, Awais Athar, and Agha Ali Raza. 2024. Uqa: Corpus for urdu question answering. *arXiv preprint arXiv:2405.01458*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The hindi/urdu treebank project. *Handbook of linguistic annotation*, pages 659–697.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Christian Charras and Thierry Lecroq. 2004. Handbook of exact string matching algorithms.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Ali Daud, Wahab Khan, and Dunren Che. 2017. Urdu language processing: a survey. *Artificial Intelligence Review*, 47:279–311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Samreen Kazi and Shakeel Khoja. 2021. Uquad1. 0: Development of an urdu question answering training data for machine reading comprehension.

Samreen Kazi, Shakeel Khoja, and Ali Daud. 2023. A survey of deep learning techniques for machine reading comprehension. *Artificial Intelligence Review*, 56(Suppl 2):2509–2569.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Arantxa Otegi, Aitor Agirre, Jon Ander Campos, Aitor Soroa, and Eneko Agirre. 2020. Conversational question answering in low resource scenarios: A dataset and case study for basque. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 436–442.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings* *of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382.

Mohamed Shaheen and Ahmed Magdy Ezzeldin. 2014. Arabic question answering: systems, resources, tools, and future trends. *Arabian Journal for Science and Engineering*, 39:4541–4564.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 858–867.

Human and Machine: Language Processing in Translation Tasks

Hening Wang[1], **Leixin Zhang**[2], and **Ondřej Bojar**[3]

[1]University of Tübingen, Seminar für Sprachwissenschaft
[2]University of Twente, High-tech Business and Entrepreneurship
[3]Charles University, Faculty of Mathematics and Physics, ÚFAL

Abstract

The present study analyzes the influence of linguistic factors (sentence ambiguities) and non-linguistic factors (visual cues) on online language processing in translation tasks. Moreover, it also offers an attempt at relating machine and human translation in a multimodal setting, an aspect that has received less attention before. We qualitatively evaluated translation outputs between subjects across different experimental conditions, as well as between human and machine translation processes. We observed a positive correlation between humans' reading time and models' next token prediction, with a higher similarity score for the translation of unambiguous sentences compared to translations of ambiguous sentences. We also found that a context-relevant image has a significant influence on translation updates.

1 Introduction

Translation is an important aspect of language use. A vast number of machine translation models have been developed over the last decades trying to assist and automatize this task. However, less attention has been paid to the architecture and mechanism of language processing during translation tasks and the relation between these processes in humans and in machines.

We attempt to provide a new perspective by taking translation as the task in assessing language processing and comparing human and machine processing in it. It is clear that the mechanisms are fundamentally different between the human brain and machine translation (MT) systems. However, according to the three levels of analysis proposed by Marr (1982), studying human translation processes can reveal how people handle ambiguity, context, and non-linguistic information. This knowledge can inform the development of more sophisticated and human-like MT systems at the computational level.

In real-world scenarios, human language processing is further compounded by external stimuli such as images or sounds, which can either assist, hinder or distract human comprehension. By examining how humans process sentences in real-time, we can identify strategies to improve machine translation algorithms, making them more adaptable and contextually aware. While some studies focus on specific aspects, such as using eye-tracking to evaluate MT systems (Doherty et al., 2010; Stymne et al., 2012) or using EEG to measure effort during human translation (Hansen-Schirra, 2017), it remains challenging to unravel how the working mechanisms of machines differ from those of humans and to what extent they are comparable (Wang et al., 2023; Lakretz et al., 2021).

In this study, we combined eye-tracking data to analyze human language processing and used surprisal obtained from GPT-2 to represent the processing of models. Our experiments examine language processing for both ambiguous and unambiguous sentences, presented with or without relevant visual cues. Furthermore, we compared machine translation outputs based solely on textual input and human translations performed under three different visual stimuli. We assess the potential influence of visual cues on human comprehension and evaluate whether multi-modal machine translation is necessary for reaching human-like performance in our setting. This inquiry is particularly pertinent due to the inconclusive results in integrating visual stimuli to enhance machine translation (Specia et al., 2016; Elliott, 2018; Caglayan et al., 2019). This attempt also allows us to relate human cognitive processes to artificial systems in future research.

The objective of the current study is to assess whether machine processing can be numerically correlated with human language processing in translation tasks. The main research question can be formulated as the following two respects:

- **Research Question 1**: Do machines and humans exhibit comparable difficulties in processing ambiguous vs. unambiguous sentences?

  Hypothesis: Higher processing complexity should be shown for both humans and machines.

- **Research Question 2**: Do visual cues impact human translation outcomes, and which visual condition in human translation aligns better with the machines' outcomes that rely solely on text?

  Hypothesis: Visual conditions affect human translation, and machine, text-only processing should be more similar to human translation results when no additional visual cues are provided.

The following sections are organized as follows: Section 2 provides an overview of previous studies on language processing and highlights the research gap in language processing, particularly in human-machine comparison that we are addressing in this study. Section 3 introduces the corpus we used in our study. Section 4 focuses on the input processing in humans and models (machine), while Section 5 analyzes the output of language processing by human and machine processing.

2 Background in Language Processing

In human language processing studies, reading time serves as a crucial measure for assessing language processing difficulty. In psycholinguistic research, there has been a comprehensive study of the correlation between processing difficulty and longer reading duration (Underwood et al., 2000; Juhasz and Rayner, 2003; Rayner and Raney, 1996). In the studies of eye-tracking techniques and language processing, fixation duration can be an indicator of processing complexity. Specifically, shorter fixation durations have been associated with more predictable words, whereas longer durations have been linked to unpredictable words (Ehrlich and Rayner, 1981).

For statistical models, surprisal theory provides a measure of the difficulty of language processing (Hale, 2001; Levy, 2008; Boston et al., 2011). Surprisal estimates how surprising or unlikely the next word appears based on the partially established structure of the sentence. For instance, the processing difficulty of garden path sentences can be captured by surprisal (Hale, 2001). In previous studies, surprisal shows a positive correlation with reading time (Smith and Levy, 2013; Monsalve et al., 2012; Goodkind and Bicknell, 2018). Roger (2008) proposes that the word surprisal is proportional to the negative log probability of words.

Another method to investigate processing difficulty can be translation output from a source language to a target language. It is found that the source text is one factor that affects translation (Campbell, 1999; Dragsted, 2012). Tokowicz and Degani (2010) state that ambiguity slows translation and can reduce translation accuracy due to the competition of potential target translation choices. Heilmann (2020) and Hvelplund (2014) study the language processing in the setting of translation tasks and state that the focus (longer gazing duration) on the source text corresponds to more translation options in the target language system. Dragsted also (2012) found that high variability of translation output is related to higher reading duration and self-corrections.

In previous studies, the complexity of language processing has rarely been examined under the human-machine comparison setting. We attempt to provide a new perspective by taking translation as the principal task in assessing language processing and comparing human and machine processing.

3 Corpus: EMMT

We use Eyetracked Multi-Modal Translation (EMMT) corpus (Bhattacharya et al., 2022) for our research. The corpus comprises 200 sentences, categorized into two types, ambiguous and unambiguous, with 100 sentences in each category.

In this corpus, source sentences are in English and they were translated into Czech. Each participant went through two rounds of reading and translating phases. In the first round, only a plain sentence was shown and the subjects were expected to say its translation into Czech aloud. In the subsequent phase, one of three visual conditions was provided: a relevant picture, an irrelevant picture, or no image. Subjects were expected to confirm their previous translation, or say an updated version. Both ambiguous and unambiguous sentences were distributed equally among the participants and across three visual conditions.

4 Input Processing

This section studies the input processing of humans and models. In Section 2, we discussed the surprisal theory and its correlation to human language processing, however, it is not yet confirmed whether surprisal also correlates with text reading specifically for translation purposes. Our study aims to fill the existing gap.

In our experiments, we also test whether an intrinsic factor (sentence ambiguity) has an impact on the language processing of humans (measured by reading duration) and the model (measured by surprisal obtained from GPT-2), and investigate whether the model's surprisal correlates with human's reading duration.

We compute the reading duration for each sentence based on eye-tracking data. The eye tracker collects data with an interval of approximately 0.5 milliseconds between each two adjacent time points. The overall reading duration of a

As the machine counterpart to human processing duration, we take the model's surprisal: the method of negative logarithm of probability proposed by Levy (2008) is adopted to compute surprisal. In addition, we view human language processing as an incremental procedure, where meaning is obtained as words are encountered in a sequential manner (Brouwer et al., 2010). Guided by this premise, we utilize the generative model GPT-2 (large) to derive word probabilities.

The probability of the next word is obtained one at a time with previous words in the sentence serving as a prompt. The predicted difficulty of a sentence is computed as the sum of negative logarithms of the conditional probabilities of the words in the sentence (excluding the first word of the sentence, which only serves as the prompt). For example, we calculate the surprisal of the sentence 'The stand is stable' as Equation (1).

$$\begin{aligned}\text{Surprisal} = &-\log(\mathrm{P}(\textit{stand} \mid \textit{The}))\\ &-\log(\mathrm{P}(\textit{is} \mid \textit{The stand}))\\ &-\log(\mathrm{P}(\textit{stable.} \mid \textit{The stand is}))\end{aligned} \quad (1)$$

Table 1 presents the results of fixation duration in two groups (ambiguous and unambiguous) when reading source texts. The results indicate a slightly longer duration that participants dedicated to reading ambiguous sentences as opposed to unambiguous sentences. However, it is noteworthy that this difference between the two groups is not statistically significant (T-test: $p = 0.161$).

	Ambiguous	**Unambiguous**
Reading (sec)	7.637	7.334

Table 1: Reading time during sentence reading phrase.

	Ambiguous	**Unambiguous**
Surprisal value	51.21	49.56

Table 2: Sentence surprisal value obtained from GPT-2.

The surprisal values for both the ambiguous group and unambiguous group are displayed in Table 2. The table demonstrates that GPT-2 perceives ambiguous sentences to be marginally more surprising than unambiguous sentences. Similarly to human processing, the difference between the two groups is not statistically significant as indicated by T-test ($p = 0.162$).

We further analyze the correlation between sentence reading duration and surprisal using Pearson's correlation coefficient (r). The results indicate a moderate positive correlation between the two ($r = 0.507$). Analyzing the ambiguous and unambiguous sentence groups individually, we find correlations of $r = 0.58$ for the unambiguous group is higher than $r = 0.43$ for the ambiguous group. This suggests that the alignment between human reading time and the model's surprisal is more pronounced in the case of unambiguous sentences.

5 Translation Outputs

In this section, we analyze the translation outputs as the results of language processing. Three experiments were implemented to investigate two factors (ambiguity and visual cues): (1) a comparison between the initial translation and subsequent updated version by the same subjects (Section 5.1); (2) a comparison of the translation outputs across different subjects (Section 5.2); and (3) a comparison of the translation outputs between humans and machine translation systems (Section 5.3).

On the one hand, we explore the effect of sentence ambiguity on translation outputs. Our study builds on previous research (Tokowicz and Degani, 2010; Heilmann, 2020; Hvelplund, 2014), which suggests that translation results exhibit greater variance for sentences that are more challenging to process. Our hypothesis is that ambiguous sentences can be interpreted in different ways, and as a result, their translations should undergo more updates when accompanied by an image in the second translation phase. Moreover, we anticipate that the translation outputs from humans and machines would exhibit greater dissimilarity for ambiguous sentences than for unambiguous ones.

On the other hand, we intend to analyze the influence of visual cues on human translations (Sections 5.1 and 5.2). Specifically, we aim to explore the conditions under which translation outputs demonstrate greater similarity across subjects when considering three different visual cues (a related image, no image, and an unrelated image) (Section 5.2). Additionally, we aim to identify the visual conditions under which human translations exhibit greater similarity to machine-generated translations that rely solely on textual inputs[1] (Section 5.3).

5.1 Translation Updates

This section analyzes translation updates, a comparison between subjects' initial translations (relying solely on source sentences) and their subsequent updated versions (when one of the image conditions is presented).

The similarity of sentence pairs is measured using the Levenshtein distance over words, which is further normalized into a similarity ratio using Equation (2) in order to minimize the impact of varying sentence lengths. This normalized similarity ratio ranges between 0 and 1, where 0 indicates no word overlap in the sentence pair and 1 indicates two sentences are identical. The analysis of translation updates is conducted considering two factors and 6 conditions in total: 2 [AMBIGUITY] × 3 [VISUAL CUES] setting.

$$\text{Ratio} = \frac{\text{len}(\text{Sen}_1) + \text{len}(\text{Sen}_2) - \text{distance}}{\text{len}(\text{Sen}_1) + \text{len}(\text{Sen}_2)} \tag{2}$$

We utilize a two-way ANOVA (with factor interaction considered) to assess the influence of the factors. The initial results confirm that there is no significant interaction between the two factors: AMBIGUITY and VISUAL CUES ($F = 1.385$, $p = 0.251$). The results also reveal that the difference between ambiguous and unambiguous sentences is not statistically significant ($F = 0.273$, $p = 0.251$), suggesting that sentence ambiguity has a minimal effect on translation updates during the second translation phase. We explain this by the nature of ambiguity types observed in EMMT data: depending on the source of the image and sentence, the sentences exhibit syntactic ambiguity (like "I saw a man with the telescope"), for which however the translation into Czech does not need to resolve the ambiguity, or lexical ambiguity (like "court", which is ambiguous between the court of justice and a tennis court), where however the remaining words in the sentence typically provide enough non-visual context for ambiguity resolution. In either case, there is no need to update the translation into Czech. The last common ambiguity type, gender ambiguity (male vs. female tennis player) is not very frequent and its visual resolution is often in line with the stereotypical solution chosen by the translators in the absence of other information.

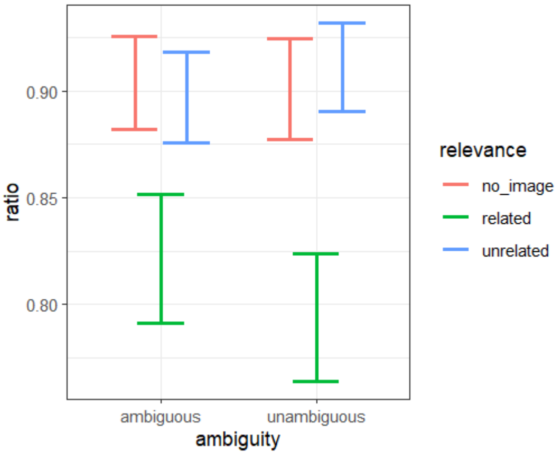


Figure 1: Similarity ratio between the initial translation and the updated translation (error bar plots).

More significantly, the test indicates a notable influence of visual cues ($F = 38.141, p < 2e^{-16}$). Figure 1 illustrates that the lowest similarity ratio occurs when a related image is provided in the second round of translation. This suggests that subjects tend to make more updates to their translation when provided with a relevant picture. Further Welch-Satterthwaite t-test shows that 'related images' exhibit a statistically significant effect on the similarity ratio of translation updates ($t = -4.588$, $p = 5.36e^{-06}$) compared to the visual condition

---

[1] Given the restriction that multi-modal machine translationreadily available, we only compare all visual conditions from humans with one condition from machines, which is only with textual input.

of ‘no images’. Our explanation here is based on the observation that the text is often *vague*. The provided image allows the translators consider the general setting in which the sentence was used, and rephrase the translation to be appropriate for this setting.

Finally, there’s no significant distinction observed between the ‘unrelated image’ condition and the ‘no image’ condition ($\beta = -0.007$, $t = -0.482$, $p = 0.630$).

5.2 Translation Comparison across Subjects

This section presents the analysis of translation similarity across subjects, specifically examining the extent to which translations of the same source sentence, produced by different subjects are similar.

Unlike the previous subsection (Section 5.1), which focuses on updates at the word or lexical level, we now evaluate the similarity of translations across subjects in terms of meaning. For this purpose, we employ the BLEURT metric (Bilingual Evaluation Understudy with Representations from Transformers, Sellam and Parikh, 2020a; Sellam et al., 2020b) to evaluate the similarity of translation pairs.

BLEURT leverages contextualized word representations from BERT to provide a score aligning better with human assessment of translation similarity (Sellam and Parikh, 2020a; Sellam et al., 2020b). The BLEURT score ranges roughly between 0 and 1, with 1 indicating more similar translation pairs and 0 less similar (the score occasionally goes below 0).

We computed BLEURT scores for all translation pairs of the same sentence across the visual conditions and subjects. More specifically, we compare sentence translations in various scenarios, such as when both subjects saw no image (written as ‘no-no’ for short); when one saw an unrelated image, and the other a related image (‘unrelated-related); etc, resulting in a total of 6 visual cues combination conditions. Overall, the study demonstrates a 2 [AMBIGUITY] × 6 [VISUAL CUE COMBINATION] setting.

Considering the repeated sampling when establishing pairwise comparisons and the potential interplay between factors, we employ a linear mixed model (with interaction and random effect structures considered) to examine the impact of ambiguity and visual conditions on cross-subject translation similarity. The linear mixed model was fitted using the REML method, and t-tests using Satterthwaite’s method.

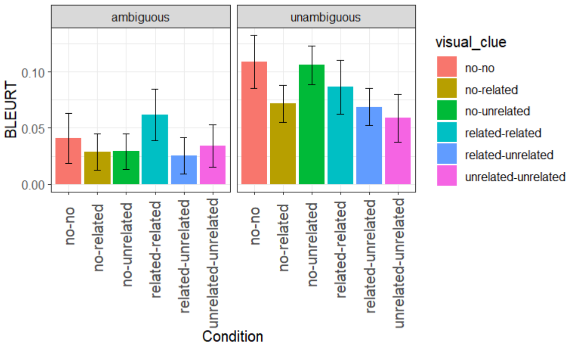


Figure 2: Cross-Subject Translation Similarity

The results are visualized in Figure 2. We observed that translations of unambiguous sentences exhibit higher cross-subject BLEURT scores than those of ambiguous sentences. It implies that unambiguous sentences are translated by humans with less variance, although this result is not statistically significant ($\beta = 6.249e^{-2}$, $t = 1.768$, $p = 0.774$).

Regarding the influence of visual conditions, the linear mixed model demonstrates that the ‘related-related’ condition is the only one demonstrating a significant effect, compared with the ‘no-no’ condition ($\beta = 4.908e^{-02}$, $t = 1.963$, $p = 0.0498$). It means that when both subjects are provided a relevant image as translation support, their translation outputs tend to be more similar compared to other visual conditions, supporting also our argument about the related image reducing the information vagueness about the described situation. This finding also implies that a relevant image may help to resolve ambiguity in the ambiguity group.

In the unambiguous group, subjects exhibited the greatest translation similarity when no image was provided for both subjects. The provision of unrelated images (‘unrelated-unrelated’ condition) results in the least similarity between translation pairs. This suggests that unrelated images might serve as distractions for subjects. However, these findings aren’t statistically significant and need further examination to verify.

5.3 Human-Machine Translation Comparison

Following the exploration of translation comparison across subjects, this section compares human translations with translations generated by four machine translation systems: Google, Lindat,[2] DeepL,

---

[2] `https://lindat.mff.cuni.cz/services/translation/`

and chatGPT.[3]

Firstly, we investigate which translation systems exhibit greater similarity to human translations. Prior research (Popel et al., 2020) suggests that the Lindat translation model (also known as CUBBITT) demonstrates higher fluency and accuracy levels than other systems, and even surpasses human translation quality. We will test the performance with our sentences and experiment settings.

Secondly, we examine visual conditions under which human translations exhibit greater similarity to machine-generated ones that rely solely on textual inputs. We use BLEURT to measure translation similarity, as in Section 5.2. Concerning ambiguity, we hypothesize that human and machine translations should exhibit greater similarity when translating unambiguous sentences. Our hypothesis regarding visual conditions is that machine translation relying solely on texts should exhibit greater similarity (higher BLEURT scores) to human translations with no images. The linear mixed model is used again (as in Section 5.2) to test the factors.

To better assess the performance of the four models, we additionally established a worst-case baseline by shuffling the Lindat translations which leads to translation pairs without association. The BLEURT score in this case is negative (-0.62). The results show that all four systems show significantly better results than the baseline ($p < 2e^{-16}$). Moreover, t-tests from the linear mix model reveal that chatGPT scores significantly lower than the other three systems ($p < 0.01$). This result might indicate a lower translation quality, but it can also be an artifact due to considerable dissimilarity between LLM-based translation outputs and standard MT outputs.

Additionally, Figure 3 demonstrates that the performance of Lindat stands out as the best among the models, although the difference from Google ($\beta = 3.63e^{-02}$, $t = 1.309$, $p = 0.191$) and DeepL ($\beta = 0.043$, $t = 1.502$, $p = 0.133$) is not statistically significant. This result verifies the performance of Lindat in prior studies.

Regarding the influence of visual cues, the t-tests conducted in the linear mixed model suggest no significant effect is observed. Nevertheless, Figure 3 provides additional insights. It shows that within the unambiguous sentence group, all four translation systems exhibit the highest BLEURT scores when their translations are compared to human translation under the 'no image' condition. This observation supports our assumption that machine translation aligns better with human translations without image assistance (though statistically insignificant, $p > 0.05$). Additionally, within the unambiguous group, translations of four systems exhibit lower BLEURT similarity scores with human translation under the condition of 'irrelevant image', compared to the condition of 'no image'. It implies that irrelevant images might distract human translators, resulting in a lower correlation between machine and human translations. However, further research is needed to confirm this hypothesis.

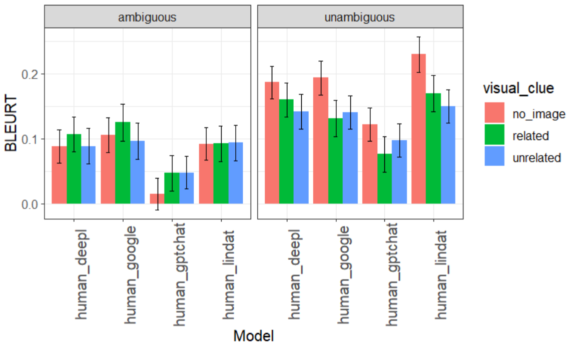


Figure 3: Similarity between human and machine translation estimated by BLEURT score taking the human translation as the reference and each of the MT outputs as candidates.

For ambiguous sentence groups, the visual conditions do not show a consistent influence on translation similarity. Translations from Google and DeepL correlate better with human translations when related images are included. However, this pattern is not apparent in Lindat and chatGPT, and the effect remains statistically insignificant.

6 Conclusion

Our study analyzes the language processing of humans and machines in translation tasks and examines the impact of sentence ambiguity and visual cues on sentence processing in translation tasks.

Section 4 suggests that processing from humans and machines correlates with each other: humans exhibit a slightly longer fixation duration, and the model reveals slightly higher surprisal values (showing higher degree of processing complexity) during the processing of ambiguous sentences.

Given the restriction that we cannot assess the machine's translation ability when providing a visual condition, we compared the machine's textual

---

[3] Translations from the respective systems were obtained in March 2023.

translation outputs with the human's translation under three visual conditions to see which condition correlates better with the machine's textual translation results. We noted that translations generated by machines tend to exhibit a higher degree of similarity to human translations when subjects are provided only with plain texts. For the unambiguous sentence group, we also observe that machine translations are more similar to human translations with only plain texts provided (without visual cues) compared to conditions with a relevant or irrelevant image.

In the examination of the effect of visual cues on human language processing, we discovered that image conditions display an influence on subjects' translation updates. In particular, when related images are provided, there is a tendency for more word updates in the later translation correction phase. In the context of translation comparison across subjects, we observed that translations tend to be more similar when both subjects are exposed to a related image. Irrelevant images might distract human translators, resulting in a lower similarity between machine and human translations.

7 Acknowledgements

This work has been supported by the GAČR EXPRO grant NEUREM3 (19-26934X), by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2023062 LINDAT/CLARIAH-CZ and by the CEDMO 2.0 NPO project.

References

Sunit Bhattacharya, Věra Kloudová, Vilém Zouhar, and Ondřej Bojar. 2022. Emmt: A simultaneous eye-tracking, 4-electrode eeg and audio corpus for multi-modal reading and translation scenarios. *arXiv preprint arXiv:2204.02905*.

Marisa Ferrara Boston, John T Hale, Shravan Vasishth, and Reinhold Kliegl. 2011. Parallel processing and sentence comprehension difficulty. *Language and Cognitive Processes*, 26(3):301–349.

Harm Brouwer, Hartmut Fitz, and John Hoeks. 2010. Modeling the noun phrase versus sentence coordination ambiguity in Dutch: Evidence from surprisal theory. In *Proceedings of the 2010 Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–80, Uppsala, Sweden. Association for Computational Linguistics.

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4159–4170, Minneapolis, Minnesota. Association for Computational Linguistics.

Stuart Campbell. 1999. A cognitive approach to source text difficulty in translation. *Target. International Journal of Translation Studies*, 11(1):33–63.

Stephen Doherty, Sharon O'Brien, and Michael Carl. 2010. Eye tracking as an MT evaluation technique. *Machine Translation*, 24(1):1–13.

Barbara Dragsted. 2012. Indicators of difficulty in translation—correlating product and process data. *Across Languages and Cultures*, 13(1):81–98.

Susan F Ehrlich and Keith Rayner. 1981. Contextual effects on word perception and eye movements during reading. *Journal of verbal learning and verbal behavior*, 20(6):641–655.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, Brussels, Belgium. Association for Computational Linguistics.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, pages 10–18.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the north american chapter of the association for computational linguistics*.

Silvia Hansen-Schirra. 2017. Eeg and universal language processing in translation. *The handbook of translation and cognition*, pages 232–247.

Arndt Heilmann. 2020. *Profiling effects of syntactic complexity in translation: a multi-method approach*. Ph.D. thesis, Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2020.

Kristian Tangsgaard Hvelplund. 2014. Eye tracking and the translation process: Reflections on the analysis and interpretation of eye-tracking data.

Barbara J Juhasz and Keith Rayner. 2003. Investigating the effects of a set of intercorrelated variables on eye fixation durations in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(6):1312.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, 213:104699.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

David Marr. 1982. Vision: A computational investigation into the human representation and processing of visual information.

Irene Fernandez Monsalve, Stefan L Frank, and Gabriella Vigliocco. 2012. Lexical surprisal as a general predictor of reading time. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 398–408.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):4381.

Keith Rayner and Gary E Raney. 1996. Eye movement control in reading and visual search: Effects of word frequency. *Psychonomic Bulletin & Review*, 3(2):245–248.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020b. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Thibault Sellam and Ankur P Parikh. 2020a. Evaluating natural language generation with bleurt. *Google AI Blog*.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Lucia Specia, Stella Frank, Khalil Sima'An, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553.

Sara Stymne, Henrik Danielsson, Sofia Bremin, Hongzhan Hu, Johanna Karlsson, Anna Prytz Lillkull, and Martin Wester. 2012. Eye Tracking as a Tool for Machine Translation Error Analysis.

Natasha Tokowicz and Tamar Degani. 2010. Translation ambiguity: Consequences for learning and processing. *Research on second language processing and parsing*, pages 281–293.

Geoffrey Underwood, Alice Binns, and Stephanie Walker. 2000. Attentional demands on the processing of neighbouring words. In *Reading as a perceptual process*, pages 247–268. Elsevier.

Shaonan Wang, Nai Ding, Nan Lin, Jiajun Zhang, and Chengqing Zong. 2023. Language cognition and language computation–human and machine language understanding. *arXiv preprint arXiv:2301.04788*.

Asking the Right Questions: Exploiting Hidden Interactions in a Generative Framework for Multilingual, Multitask Classification

Sebastian-Antonio Toma, Camelia Lemnaru, Vlad-Andrei Negru and **Rodica Potolea**
Computer Science Department, Technical University of Cluj-Napoca, Romania
`Toma.So.Sebastian@student.utcluj.ro`,
`{Camelia.Lemnaru, Vlad.Negru, Rodica.Potolea}@cs.utcluj.ro`

Abstract

This study explores the potential of leveraging additional training data as instructional prompts for a generative model in a multilingual, multitask recipe classification problem. By incorporating different tasks as additional questions, derived from data available only during fine-tuning, we aim to improve the classification performance of a sequence-to-sequence model for all tasks and languages involved. Furthermore, we investigate the impact that prompt-engineering has on the additional questions during fine-tuning, uncovering its significant role in helping the model learn hidden interactions between tasks. The proposed method produces absolute improvements of 2.3%, 6.22%, and 10.7% respectively in weighted multilingual accuracy (on three targeted classification tasks). The most effective additional actions are the questions derived from supplementary data, while the size of the model and whether we perform in-domain pre-training do not improve the final performance significantly. Our findings also underline the importance of training data selection and questioning strategies, especially in underrepresented languages, where we obtained an absolute increase in accuracy of 34.8% in the few-shot setting and 30.33% in the 0-shot setting for an underrepresented language in a difficult main task, together with an increase from 0% to 97% in F1-score for the most underrepresented class.

1 Introduction

Text classification has become increasingly important for effectively analyzing vast amounts of textual data across different languages, in the context of the diverse and growing multilingual landscape of digital content (Li et al., 2021). It can also be applied in a Multi-Task Learning (MTL) setup, with the aim to improve the performance and efficiency of Natural Language Processing (NLP) models by simultaneously learning multiple, related tasks (Hupkes et al., 2023).

This paper focuses on the application of MTL techniques in the context of multilingual text classification, with the declared goal of leveraging the inherent relationships between different classification tasks to improve the accuracy and robustness of the model used. We employ a multilingual generative model as the backbone and focus on three classification tasks, where the labels represent cooking properties for oven recipes. In addition to the labels for the three target tasks, the multilingual data contains annotations related to other properties of the dishes, such as dish type, size, certain ingredients, or oven settings. The proposed framework effectively captures both the information shared between the three target tasks, but also capitalizes on the additional annotations (available only during fine-tuning), by introducing new tasks through the use of instructional prompts (or questions). Thus, the model is able to explicitly learn from the synergies between the new tasks and the target tasks during the fine-tuning phase.

The main contribution of the paper is to propose an instruction-driven, joint learning generative framework that helps the model extract hidden correlations for a better classification of recipes, especially for underrepresented languages and classes, in a highly imbalanced dataset. Unlike Wu et al. (2022), from which our approach is inspired, our study generates the content of these instructional prompts (or questions) from auxiliary annotated sparse data available only during training, the model being charged with understanding and predicting the respective answers.

Additionally, we:

- perform an ablation study on the selection of the most relevant annotations to use to generate instructions, also guided by their correlation with the target problems, obtaining an increase in the overall accuracy of 2.3% in the few-shot setting and 5.22% in 0-shot and 15.56% 0-shot for an underrepresented language.

- explore the benefit of in-domain pre-training, which produced some improvements for some of the problems and languages, but no consistent behavior was observed.
- perform evaluations in several fine-tuning scenarios: 0-shot vs. few-shot evaluations, using underrepresented vs. well-represented languages only, and using various sizes of the backbone language model, obtaining increases in absolute weighted accuracy of 2.3%, 6.22%. and 10.7% per problem, and 34.8% few-shot and 30.33% 0-shot for a difficult problem in an underrepresented language.

Finally, we obtain an improvement in the F1-score for some of the lowest sampled classes, in the underrepresented languages, from 0% to 97%, proving classification on highly imbalanced datasets can benefit from our method.

2 Related work

The idea of jointly training a single model on multiple tasks to enable the sharing of knowledge and representations across tasks has been explored in various NLP applications, from intent detection and slot filling, to joint entity classification, relation classification, and co-reference clusters in scientific literature, or machine translation (Chen et al., 2021). Lăpuşan et al. (2022) apply such an approach on BERT (Devlin et al., 2019) and other variations (RoBERTa, Liu et al., 2019), to perform German Cooking Recipe Classification on four labels related to oven parameters. They obtained the best classification performance using the domain-adapted pre-trained language model on the recipe title concatenated with the instructions, in a joint training regime.

The advent of generative models has opened up a new range of possibilities for developing mechanisms that efficiently exploit the compositional capabilities of language models through prompting, either via fine-tuning, in-context learning or even augmenting them with reasoning skills and external tools (Mialon et al., 2023, Al-Negheimish et al., 2021). Liu et al. (2021) present a novel paradigm that shows how prompt-engineering can be used to discard the pre-train and fine-tune approach of Large Language Models and replace it with "pre-train, prompt, and predict" methodology, using pre-defined prompts to reveal the model prior knowledge. Wu et al. (2022) introduce a Unified Generative Framework (UGEN) to model the tasks as question-answering problems for joint multiple Intent Detection and Slot Filling. They use a template of five questions during training to extract relevant information from the context, such as keywords, that will help the model to better generalize during the evaluation phase, where only two out of five questions were used (without the auxiliary helping questions used during training). Chain of Thought Prompting (CoT, Wei et al., 2023) explores the emergence of complex reasoning capabilities in LLMs via prompting in a sequence of carefully selected demonstrations.

3 The data

Our target is to improve multi-task, multi-lingual classification behavior using additional data available only during training, by using the MTL paradigm. To accomplish this, we used a private dataset in the cooking domain having these additional annotations and some difficult main tasks that can benefit from this method. It consists of tens of thousands of online cooking recipes scrapped from web (including a wide range of **public** cooking websites), in 6 languages: English, German, Dutch, Italian, French, and Swedish. German and English are overrepresented in the labeled data, while Italian and Swedish are the most underrepresented. Each recipe has been annotated for three target tasks:

- Meat-centric (**M.**) - binary text label with "yes" or "no" classes, representing whether the main "focus" of the dish is (a loaf of) meat
- Surface aspect level (**S.l.**) - binary numeric label: classes "1" & "2", representing the cooked product's aspect/color
- Dehydration level (**D.l.**) - label with 5 numeric classes: from "1" to "5", representing the moisture reduction degree.

Additionally, besides the main classification tasks of **M.**, **S.l.**, and **D.l.**, each recipe has:

- Title
- Instructions
- Oven settings

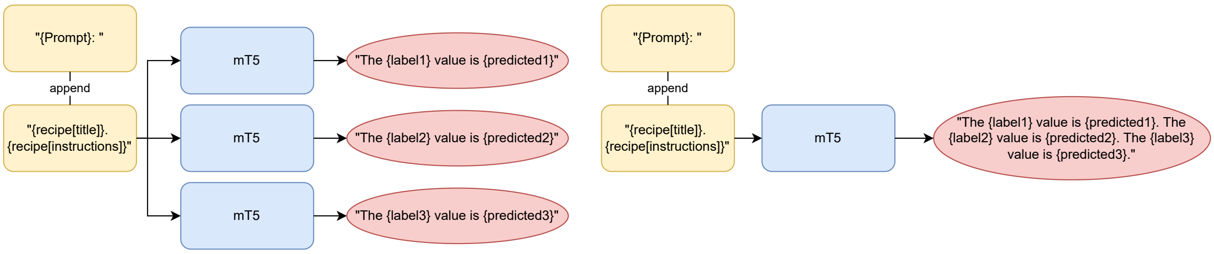

"{Prompt}: "
append
"{recipe[title]}.
{recipe[instructions]}"
mT5
mT5
mT5
"The {label1} value is {predicted1}"
"The {label2} value is {predicted2}"
"The {label3} value is {predicted3}"
"{Prompt}: "
append
"{recipe[title]}.
{recipe[instructions]}"
mT5
"The {label1} value is {predicted1}. The {label2} value is {predicted2}. The {label3} value is {predicted3}."

Figure 1: Architecture of the baseline model (left) vs. architecture of the joint model (right).

Initial setting	**Final setting**	**Description**
$a \vee b$	b	"OR" operands with missing/neutral values are discarded
$d \vee b$	b	
$c \vee b$	c	If both have missing/neutral values or both complete, take first

Table 1: Pre-processing procedure of multi-step recipes' oven settings, where a = *"T: 200°C, t: 25min, P: Not known"*, b = *"T: 100°C, t: 25min, P: Gas"*, c = *"T: 212°F, t: 25min, P: Grill"* and d = *"T: null, t: 25min, P: Gas"*.

- Some additional annotations related to the required cooking settings, dish type, certain ingredients, size, type, thickness, etc.

For each recipe, the default text used by the model as input consists of the recipe's **title** and **instructions** extracted from the HTML content, as in Lăpuşan et al. (2022).

4 Proposed Method

The selected **Baseline** model uses the default input (*title+instructions*) to predict only one of the three main tasks (**M.** or **S.l.** or **D.l.**), resulting in three baseline models, one for each.

Next, each incremental step is described together with the name used for the model and the part of the dataset being added.

4.1 Joint Learning

To better exploit the hidden correlations between the target problems, we first trained a **Joint** model, fine-tuned on all three tasks combined into one. The chosen order of generation is from the easiest task to the hardest task (as indicated by baseline, individual models: lower accuracy obtained for the task, means for us that the task is more difficult): $M. \rightarrow S.l. \rightarrow D.l.$. This way, the latter predicted labels should benefit from already having available the labels predicted before. This is one of the reasons we employed a sequence-to-sequence generative model for our classification tasks, our main focus being the **S.l.** and **D.l.** tasks, as for these the baseline model seems to struggle more. The difference between the architecture of the baseline model and the joint model is illustrated in Figure 1.

4.2 Additional Fine-tuning Task

Although not among the target tasks, we considered using the oven settings as an additional task in the joint model, only during fine-tuning, with the highest precedence compared to the target tasks: $OvenSettings \rightarrow M. \rightarrow S.l. \rightarrow D.l.$ The intuition was that the oven settings should influence the most the outcome (especially for **D.l.** and **S.l.** tasks, which should also be affected by whether the meat is the main content of the dish). The additional task of predicting the oven settings is formulated as predicting the triplet (T, t, P):

- Temperature setting (either in ^oC or ^oF)
- Time setting (in minutes)
- Cooking Program

Some of the recipes in the available dataset contain multiple such triplets, either because they are multi-step (in which case the triplets are joined with "AND"), or because the recipe provides alternative cooking instructions (pairs of triplets joined by an "OR"), we applied a pre-processing procedure, keeping at most five steps and only one of the options in each disjunction (see Table 1 for details):

$$\bigwedge_i (a_i \vee b_i) = \bigwedge_j c_j, j = min(i, 5)$$

where i is the number of steps.

This procedure was applied to reduce the number of tokens used for this task, to not use too many

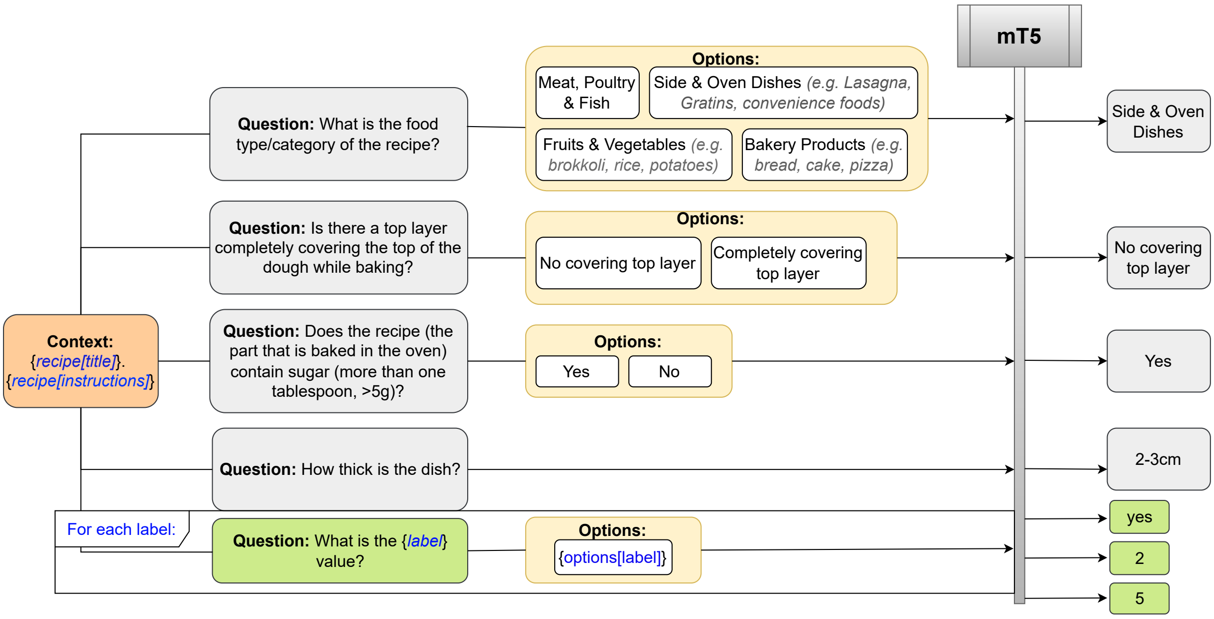

mT5
Context: {recipe[title]}. {recipe[instructions]}
Question: What is the food type/category of the recipe?
Options:
Meat, Poultry & Fish
Side & Oven Dishes (e.g. Lasagna, Gratins, convenience foods)
Fruits & Vegetables (e.g. brokkoli, rice, potatoes)
Bakery Products (e.g. bread, cake, pizza)
Side & Oven Dishes
Question: Is there a top layer completely covering the top of the dough while baking?
Options:
No covering top layer
Completely covering top layer
No covering top layer
Question: Does the recipe (the part that is baked in the oven) contain sugar (more than one tablespoon, >5g)?
Options:
Yes
No
Yes
Question: How thick is the dish?
2-3cm
For each label:
Question: What is the {label} value?
Options:
{options[label]}
yes
2
5

Figure 2: Question-driven Generative Framework with an example of question templates used only for training (gray boxes/first 4) and questions for the main tasks (**M.**, **S.l.** and **D.l.**) used for testing (green box/last one), with the corresponding options, and their explanatory information (gray text in parentheses).

tokens from the main tasks. We reduced the number of tokens used for the output of this task to 113 maximum. The percentage of entries in this dataset affected are (per language): De 9%, En 57%, Nl 1%, Fr 7%, It 15%, Sv 3%.

This model would be referred to as **OvenSettings**.

4.3 Question-driven Generative Framework (QdGF)

The central part of our approach is the Question-driven Generative Framework (**QdGF**), which uses a template to generate different types of questions from auxiliary/redundant training data. Inspired from Wu et al. (2022), our approach generates the response to the questions from auxiliary data available only during training and does not extract it from the target labels. To this extent, the model is fine-tuned in a multi-task manner, but with a variable number of tasks, as some of the additional information is not available for all the recipes in the training data. The only tasks to be guaranteed for all recipes are the main tasks, transformed in the same manner as the other tasks, to questions. The data is constructed in the following format: "<s>Context: $\{recipe[title]\}$. $\{recipe[instructions]\}$</s> Question: $\{Q_i\}$? Options: $\{options_i + explanations_i\}$</s>".

The number of questions/instructional prompts (i) used ranges between 6 and 10 per experiment. All these questions represent a subset from a set of 14 extracted questions initially. That is, we concatenate more information from the dataset (the recipe's title and instructions, together with several questions from the set of used questions, and their **answer options**, provided that the recipe is labeled with those annotations in the dataset), and use the resulting string as input to the mT5 model in the fine-tuning stage.

With these new prompts added, we try to stimulate the prior general knowledge of the model (Han et al., 2021) and direct it to a greater focus on this downstream task composed of the 3 target problems (**M.**, **S.l.**, and **D.l.**). Several elements in the options list also contain additional explanations that help guide the model, but these were removed from the labeled option. Providing the options to choose from and some related explanations (where available) should ground the model to the current problem domain and prevent potential hallucinations (Ahn et al., 2022).

A discussion on how the subset of used questions was chosen from the main set of questions is presented in Section 5.5 and an example of a possible instantiation of the framework is illustrated in Figure 2 (the best architecture used is similar to this, but with joint training for the three main tasks).

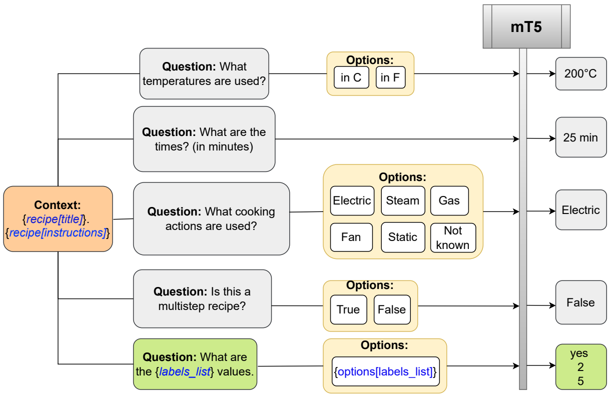

mT5
Context: {recipe[title]}. {recipe[instructions]}
Question: What temperatures are used?
Options:
in C
in F
200°C
Question: What are the times? (in minutes)
25 min
Question: What cooking actions are used?
Options:
Electric
Steam
Gas
Fan
Static
Not known
Electric
Question: Is this a multistep recipe?
Options:
True
False
False
Question: What are the {labels_list} values.
Options:
{options[labels_list]}
yes
2
5

Figure 3: QdGF instantiated with the subset of oven settings questions.

Lang.	M.	S.l.	D.l.
German	12:1	2:1	12:6:4:2:1
English	9:1	2:1	10:6:5:1:1
Dutch	7:1	1:1	10:8:8:3:1
French	12:1	2:1	4:2:2:1:1
Italian	17:1	2:1	7:4:4:4:1
Swedish	13:1	2:1	7:6:4:3:1
38:16:8:4:3:1	10:1	2:1	8:5:3:1:1

Table 2: Imbalance ratios per language (first column, last row), imbalance ratios per language and per task (with 2, 2, and 5 classes respectively) and imbalance ratios per task (last row, last three columns).

4.3.1 QdGF + OvenSettings

We transformed the triplet (T, t, P), from Section 4.2, into 3 more questions, one for each oven setting, to further exploit the previous approach and adapt it to this framework. We also added an additional question, to make a difference between multi-step recipes (having more such triplets) and single-step recipes (with just one triplet). An example framework with this subset of questions is illustrated in Figure 3. The total number of potential additional questions to choose from, including the oven settings-related questions, is now 18.

All the extracted questions are available in Appendix B.

4.3.2 Renaming

One more potential issue we tried to overcome in this framework is related to the use of numbers to encode the class labels, which bears either no, or a potentially wrong semantic meaning for the model (Spokoyny et al., 2022). To address it, we changed each class from its numeric counterpart to a textual description consisting of 1-3 words, which we considered to best capture the class meaning. For example, for **S.l.**, the resulting mapping to textual classes is: $1 \rightarrow Maillard$; $2 \rightarrow Caramelization$. The full mappings can be found in Appendix A. We integrated these newly renamed labels directly into QdGF, with their respective explanations in the **Options** component.

5 Experiments and Results

The dataset used contains approximately 52000 multilingual recipes labeled with the three target tasks and several additional annotations (few of them being sparse), performed by human experts (see Section 3). It is highly imbalanced both respective to the languages and the classes used (see Table 2).

The train-validation-test split is $0.64 - 0.20 - 0.16$ for German and English (the fine-tuning languages). We fine-tuned only with the high-resource languages. The other languages are used in a 0-shot setting and also in a few-shot similar setting, due to the limited samples (Dutch 6%, French 2.6%, Italian 1.9%, Swedish 0.8%-shot setting respectively).

The backbone model employed is mT5 (Xue et al., 2020), a multilingual encoder-decoder transformer, and an attention-based model (Vaswani et al., 2017). Through manual hyperparameter tuning, we found the following hyperparameter values to produce the best results (and were used further): *Learning rate:* $3e-4$, *Weight decay:* 0.1, *No warmup* with Optimizer *AdaFactor* and Scheduler: *AdafactorSchedule*. Initially, we used AdamW Optimizer (Loshchilov and Hutter, 2017) and the linear scheduler with warmup from HuggingFace, but we changed to the original optimizer used for the T5 model (AdaFactor, Shazeer and Stern, 2018) which produced better results.

5.1 Specialization Pre-training

Initially, for all approaches, we used the mT5 model pre-trained by Google[1] on mC4. To further "specialize" the model on culinary data, we decided to further pre-train it on a multilingual dataset of approximately 490000 unlabeled recipes. As this dataset was not comparable in size to the original training corpus, pre-training the model from scratch using only these recipes did not provide enough information for the model to acquire a general semantic knowledge of words. Hence,

---

[1] `https://huggingface.co/google/mt5-base`

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	95.82	94.61	91.65	91.41	94.17	91.47	94.68
	large	97.26	96.83	92.57	95.91	95.63	95	96.43
Joint	large	97.3	**97.09**	93.78	95	**98.43**	93.38	96.7
Pt Joint	base	96.83	96.36	**95.51**	**96.59**	**97.81**	**96.69**	96.59
QdGF*	large	97.92	96.67	**94.46**	97.06	96.81	95.33	**96.98**
Pt QdGF*	base	**98.21**	**96.78**	93.97	96.56	95.84	96.33	**96.98**
	large	**97.96**	96.52	93.88	**97.17**	96.95	**97**	96.97

Table 3: **M.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	88.65	86.76	80.43	57.82	64.48	72.5	84.44
	large	90.88	89.62	71.25	70.68	70.4	53.33	85.87
Pt Joint	base	91.08	91.33	86.75	89.09	**90**	78.51	90.3
QdGF*	base	90.93	**92.94**	86.63	87.26	87.1	**88**	90.03
	large	90.15	91.22	**86.22**	**89.89**	**85.99**	85.66	89.39
Pt QdGF*	base	**91.86**	92.52	87.04	89.38	87.38	87.66	**90.66**
	large	**91.17**	**91.95**	85.82	87.56	85.57	**87.33**	**89.74**
Pt QdGF renam.*	base	91.28	90.75	**89.08**	**90.18**	87.81	87.33	90.4

Table 4: **S.I.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

the specialization pre-training was performed on the already pre-trained on mC4 model. We considered different approaches of pre-training, between Masked Language Modelling and Next Sentence Prediction, as Sun et al. (2022) highlights the reintroduction of NSP as a pre-training approach and also the importance of the pre-training corpus. However, we choose the original pre-training approach of the T5 model (Text-To-Text Transfer Transformer, Raffel et al., 2020), using a script performing span-masked language modeling[2].

5.2 Size Does(n't) Matter?

We tested all the sizes of mT5 that we were able to fit within our resources: *small*, *base*, and *large*, but focusing only on *base* and *large* as *small* did not stand up to the complexity of the tasks. For the first methods, *large* seemed to perform slightly better than *base* overall, but most importantly in the few-shot and 0-shot setting for the underrepresented languages. With the approach of incorporating the QdGF though, *base* performed similarly, or even slightly better than *large*, hinting that exploitation of hidden correlations using the right questions might be more important than the size and number of parameters of the model. Maybe this framework reduces the need for memorization, where large language models tend to be better (Tirumala et al., 2022). However, this claim is limited by the fact that, as the architecture grew more complex, the input data needed to be truncated for the *large* model to fit into available memory, thus maybe affecting performance. More on this in Limitations. We emphasize that our goal was not to compare between the two sizes, but to observe the improvements our method brings to each model (size) independently.

5.3 Overall Results

For assessing results, we considered the best models per problem out of the following (trained on De & En, few-shot setting for Nl, Fr, It, Sv):

- **Joint** (all three problems at once, fine-tuned on all languages, see Section 4.1).
- **OvenSettings** (joint model with the additional fine-tuning task, see Section 4.2).
- **QdGF** (model with a Question-driven Generative Framework, see Section 4.3), only the best instantiation of the questions is reported in the final results. Results from other instantiations are discussed in Section 5.5.

---

[2] `https://github.com/huggingface/transformers/blob/main/examples/flax/language-modeling/run_t5_mlm_flax.py`

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	79.54	73.46	66.8	60.35	55.38	59.69	74.28
	large	86.51	86.71	68.4	64.55	57.01	59.17	81.71
Pt Joint	base	87.17	86.5	82.26	74.54	72.18	72.72	**84.98**
Pt OvenSettings*	base	87.28	83.75	**82.77**	**78.91**	**74.45**	**86.44**	**84.98**
	large	86.98	84.58	**82.34**	**80.04**	**73.21**	**85.59**	**84.96**
Pt QdGF*	base	87.37	**87.28**	81.25	75.12	72.54	81.66	84.04
Pt QdGF+OvenSettings*	base	**87.77**	86.13	81.25	75.22	70.87	82	83.9
	large	**87.68**	**87.12**	80.93	77.65	68.51	78.66	83.98

Table 5: **D.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv
Baseline	base	88 / 84	91 / **90**	22 / 84	0 / 78	0 / 55	0 / 60
	large	89 / 86	94 / **92**	34 / 85	0 / 81	0 / 53	0 / 50
Pt OvenSettings*	base	**91** / 81	**95** / 81	**90** / 85	**61** / **85**	**76** / 55	**97** / **83**
	large	90 / 82	93 / 83	**90** / 86	**63** / **87**	**75** / **61**	94 / **91**
Pt QdGF+OvenSettings*	base	**91** / **86**	84 / 89	**90** / **87**	55 / 83	70 / **67**	94 / 76
	large	**91** / **88**	**98** / 88	83 / **88**	49 / 85	66 / 59	**97** / 75

Table 6: **D.l.** - F1-scores for the 2 most underrepresented classes out of the 5 classes (2ndUnderrepClass / mostUnderrepClass). * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

- **QdGF+OvenSettings** (the model with a Question-driven Generative Framework that includes the question obtained from the additional task, see 4.3.1).

- **QdGF renam.** (the model with the renaming of classes for **S.l.** and **D.l.** tasks, see 4.3.2).

The overall results were considered per language, and for all languages combined using the weighted accuracy (with respect to the number of samples of each language). We also considered a mix of joint or not joint and specialized pre-trained or just default pre-trained. The baseline used to compare our results is one model per problem fine-tuned on all languages with a simple instructive prompt (dubbed "Baseline", see Section 3). We report the average result of multiple runs (with standard deviations up to 0.1%) for every mentioned model.

The final results for **M.** can be seen in Table 3. The best models based on the weighted accuracy are both QdGF Joint, one specialized pre-trained (with an increase of 2.3% compared to *base* Baseline) and one not. Based on the accuracy on the underrepresented languages, the specialized pre-trained Joint model performs the best, with an increase of 5.22% for Swedish (in a *base* model). We did not expect the renaming to help for this task since it was not applied to the classes of this problem (the classes already being "yes" or "no").

For **S.l.** (Table 4), the highest accuracy has been obtained by specialized QdGF joint again (increase of 6.22% in base). In this task, renaming (the second best) helped a lot, especially for the French language, with an increase of 32.36%. Also, the specialized joint model performed well (especially in Italian, with an increase of 25.52%), proving that this problem benefits from being joined with **M.**.

We expected oven settings to be related to **D.l.**, therefore the best models are the ones involving these auxiliary settings, as can be seen in Table 5. The specialized OvenSettings Joint model performed the best in both *base* and *large* and both, weighted accuracy and accuracy of every underrepresented language. This task also benefits from being joined with the other previous tasks. Specialized QdGF joint, which is the best model for the other two problems, is also one of the best here, followed by its extension with the settings auxiliary questions in weighted accuracy (with its German and English results being one of the best).

The QdGF approaches also helped to recognize the lowest represented classes, as the dataset is highly imbalanced, especially for **D.l.** task (Table 2, last column). For this, we measured the F1-score for the 2 most underrepresented classes in this problem, and the comparison can be seen in

Task	thickness	top	appearance	type	cheese	sugar	dough	dishes	temp
S.l.	-0.02	-0.03	0.17	-0.21	**1.00**	**1.00**	-0.49	-0.21	-0.10
D.l.	0.05	0.75	-0.84	-0.20	-0.01	0.25	**-1.00**	**1.00**	0.01

Table 7: Relevant correlations between **S.l.** and **D.l.** and answers of selected questions (Pearson correlation was used).

Table 6. A reason for the performance discrepancies in Table 6 - *Baseline row* between the second most underrepresented class (with an F1-score of 0 for underrepresented languages) and the most underrepresented class (with F1-score > 50) might be the model overfits on De & En (thus a higher F1-score for that class for De & En) and is unable to generalize to the other languages. We can see that employing Joint learning with OvenSettings and even QdGF, reduces the overfitting significantly.

Initially, we tested all the models (trained on De & En) in the 0-shot setting for the underrepresented languages (Nl, Fr, It, Sv). The model performed a bit better in these underrepresented languages after seeing examples from them. However, in some cases, it performed slightly worse in De & En, but nothing significant (in some cases it even performed better), implying that no catastrophic forgetting took place and that our method is able to generalize well even in new languages. These evaluations, together with the complete evaluations in the few-shot setting can be seen in Appendix C.

5.4 Domain generalizability

The recipe data used exhibit significant diversity in terms of structure and format, which means the model may not simply learn the specifics of the cooking recipes themselves. This, and the fact that specialization pre-training in the cooking domain did not help that much, suggests that the method can be generalized to other domains, by instantiating a QdGF with questions obtained from additional/redundant data, extracted data from the text to classify, or even metadata available in a new dataset/domain.

5.5 Ablation Study

The set of 18 available questions (see Appendix B) would result in 2^{18} possibilities of instantiations of the QdGF. To select the subset of questions, we checked for the most sparse questions among the dataset to try to avoid them, and we checked the correlations of the answers to the questions with our three main problems. We can see in Table 7 that the most correlated types of questions with the **S.l.** problem are *cheese* and *sugar*, which are contained in the best model for this problem (which is a **Pt QdGF*** with final acc. 90.66%). A decrease of 2.3% occurs in weighted accuracy few-shot, 5.22% in 0-shot, and 15.56% 0-shot for an underrepresented language, if discarding one of them. For **D.l.**, the most correlated questions (*dough*, *dishes*) are also part of the best QdGF model for this problem (Pt QdGF* with 84.04% and this one+OvenSettings* with 83.9%). The best versions of QdGF models in the final evaluations (Appendix C) differ between **S.l.** and **D.l.** by the subset of questions used, only the best were presented for each. All chosen subsets contain the *type* question as this correlates approx. -0.20 with **S.l.** and **D.l.** and 0.32 with **M.**.

6 Conclusion

Our study demonstrates the effectiveness of leveraging additional training data as instructional prompts in a multilingual, multitask classification problem, by introducing QdGF (Question-driven Generative Framework). Our proposed method achieves notable improvements in weighted multilingual accuracy, with absolute improvements of 2.3%, 6.22%, and 10.7% for the targeted classification tasks. Notably, the additional tasks related to oven settings and the highly correlated ones with the specific problems have the most significant impact. We observe that the size of the model and in-domain pre-training have minimal impact on final performance. Our findings underscore the importance of thoughtful training data selection and questioning strategies, particularly in underrepresented languages and imbalanced datasets. In such cases, we achieved substantial accuracy increases of 34.8% in the few-shot setting and 30.33% in the 0-shot setting, and a 97% increase in the F1-score of underrepresented classes, for the most underrepresented language. These results highlight the potential of leveraging additional training data and prompt-engineering to improve performance on multilingual, multitask models in text classifications.

Limitations

Truncation

Due to limited resources (a system with multiple GPUs NVIDIA Tesla V100 SXM2 16 GB, from which we used on average 3 GPUs per run), we were unable to run larger models than *mt5-large* (1,2B+ parameters). Regarding the maximum number of tokens accepted, for *base*, we used 512 tokens. Thus, recipes with more than 512 tokens were truncated. Approximately 500/55000 recipes were affected. As the architecture grew more complex, *large* became not as reliable as *base* due to the extra truncation needed. For the *large* model, we had to set the maximum number of tokens to 320, approx. 2600 recipes being affected, and for the more complex approaches (such as QdGF), we had to set the maximum number of tokens to 254, 5100+ recipes being affected, thus maybe affecting performance (as we can see in the latest tables, that results start to drop for *large* models as they get more complex). We did not test specifically how much truncation affects performance. Our main focus was the *base* model and we did not want to interfere with its truncation so we can measure this method's performance increase. A deeper study on what is the optimal truncation could be done in future. Setting the tokens to the same size to compare *base* vs *large*, to see which one generalizes better with our method, constitutes another future research interest, as this focused only on the improvements brought by the method to each model (size) individually. Therefore, access to more resources/better memory usage would make our Framework scalable to longer text also (longer than what the memory can fit, per model size), as it would not need this truncation.

Backbone model

We did not use other encoder-decoder models because we wanted to highlight the improvement this method brings to a language model, and we chose to show it on mT5, thus we selected mT5 (without the QdGF framework) as a baseline. Having other models as baselines constitutes another research interest of ours, depending on the available resources. The high computational cost and resources associated to larger and newer models were the main reasons we were unable to test our method on such other models. When it comes to LLM APIs, most of the services incur additional costs for fine-tuning, that were not available to our study. Moreover, this would raise concerns about the security and privacy of our data, since the dataset used is private.

Computational expense

Another resource-related limitation would be that our framework is computationally expensive, as, for every question, we replicate the recipe for that question (only if an answer is available). For a framework with 6 questions, the training data can grow up to 6 times, per epoch. This also increases the computation time proportionally. To solve this, we could take advantage of the context of LLMs, by prompting first the recipe, followed by the questions, without replicating the recipe each time. This would require a larger model. Another solution would be to feed the model a larger prompt composed of the recipe and all questions plus the desired tasks at once, but this again has the same downsize as mentioned before. Given the size of the output, this can also be susceptible to hallucinations, which can impact performance.

Data

Our method proved to work for: German, English, Dutch, French, Italian, and Swedish. This method might not work in languages not supported by the multilingual model. For new underrepresented languages (supported by mT5) our model will bring significant improvements in a 0-shot setting as our experiments showed. For the few-shot setting though, some additional annotations might need to be added/extracted manually for maximum improvement, which requires additional labor (although the effort might not be worth it, as results in the 0-shot setting are not much smaller).

We discovered a few wrongly labeled recipes in the dataset and a few recipes not properly scrapped from Web (containing HTML tags in instructions). We solved them, but we cannot guarantee that the final dataset was 100% clean and with no noise that might have affected performance, but we can say that QdGF adds some robustness to such noise, as the model "answers" some questions before making a final decision.

Our method is tested currently only on one unique dataset. We would like to emphasize that the dataset we used is relevant enough, both in terms of the amount of data, languages, and additional questions, especially for fine-tuning and 0-shot scenarios. However, the dataset is specific to the cooking domain. Applying a QdGF approach to data and problems from different domains might need data pre-processing and extraction of questions.

Ethics Statement

An important ethical concern is high energy consumption. As discussed in Limitations, our method requires computational resources that are avid energy consumers. A significant amount of electricity was consumed for running experiments, taking on average 3 hours per fine-tuning experiment (on 3 GPUs) and 12 hours per pre-training (on 4 GPUs).

Another ethical concern is the potential bias that may be present in the training data used to fine-tune the model, as the data were scrapped from public websites. It is essential to ensure that the dataset is free from misleading content.

Additionally, when deploying the model in real-world applications, it is crucial to consider the impact of classification errors or misinterpretations, particularly in kitchen safety domains, as various oven settings are involved.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances.

Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. 2021. Numerical reasoning in machine reading comprehension tasks: are we there yet?

Shijie Chen, Yu Zhang, and Qiang Yang. 2021. Multi-task learning in natural language processing: An overview.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. State-of-the-art generalisation research in nlp: A taxonomy and review.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2021. A survey on text classification: From shallow to deep learning.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alex-Mihai Lăpuşan, Rareş-Liviu Horge, Sara Petres, Mihaela Dînşoreanu, Rodica Potolea, and Camelia Lemnaru. 2022. Instructions are all you need: Cooking parameters classification for monolingual recipes. In *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 73–80.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

Daniel Spokoyny, Chien-Sheng Wu, and Caiming Xiong. 2022. Numerical correlation in text. In *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 33–39, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. Nsp-bert: A prompt-based few-shot learner through an original pre-training task–next sentence prediction.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. 2022. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7203–7208, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

Appendix

A Renamings

A.1 Surface aspect level (S.l.)

- $1 \rightarrow Maillard$
- $2 \rightarrow Caramelization.$

A.2 Dehydration level (D.l.)

- $1 \rightarrow Maintain\ elasticity$
- $2 \rightarrow Not\ too\ moist$
- $3 \rightarrow Rising\ dough$
- $4 \rightarrow Steam,\ dry,\ grill$
- $5 \rightarrow Crispy\ bottom.$

B The set of Questions

Each question is identified by its ID. Some answer options might contain additional explications (enclosed in parentheses).

B.1 Top

Is there a top layer completely covering the top of the dough while baking? *Options*:

- No covering top layer
- Completely covering top layer

B.2 Browned

Shall your food be browned on top? *Options*:

- Yes
- No

B.3 Appearance

How would you describe the appearance of the food? *Options*:

- One large
  (e.g. Lasagna, gratins, casseroles, etc.)
- Few thick items
  (grilled, stuffed foods)
- Many small items
  (e.g. french fries)
- One large thin item with crispy bottom
  (e.g. Quiche, Pizza, Tarte)

B.4 Prepare

How do you want to prepare the food? *Options*:

- Roasting one or few large pieces with a crispy surface
  (e.g. whole chicken or a roast)
- Cooking and baking of a casserole in a large container
- Cheese on top
- Airfrying many smaller pieces with a crispy surface
  (e.g. chicken legs, chicken wings)

B.5 Marinade

Do you use a sweet rub or marinade (e.g. honey rub or rub with brown sugar)? *Options*:

- Yes
- No

B.6 Cheese

Does the recipe use one of the following cheeses (Mozzarella, Pizza cheese, Gratin cheese)? *Options*:

- Yes
- No

B.7 Sugar

Does the recipe (the part that is baked in the oven) contain sugar (more than one tablespoon, >5g)? *Options*:

- Yes
- No

B.8 Dough

What type of dough/batter is it? *Options*:

- Any other
  (e.g with baking powder, not sure)
- Yeast and Bread doughs
- Puff Pastry

B.9 Dishes

Is it one of the three dishes: Pizza, Quiche(s) or Tarte(s)? *Options*:

- Yes
- No

B.10 Thick

How thick is the dish? Consider only the part placed in the oven. *Options*: in cm.

B.11 Pastry

Is the puff pastry filled with meat or fish? *Options*:

- Yes
- No

B.12 Type

What is the food type/category of the recipe? *Options*:

- Bakery Products
  (e.g. bread, cake, pizza)
- Side & Oven Dishes
  (e.g. Lasagna, Gratins, convenience foods)
- Fruits & Vegetables
  (e.g. broccoli, rice, potatoes, etc.)
- Meat, Poultry & Fish
- Not sure

B.13 Preheat

Does it need preheating? *Options*:

- Yes
- No

B.14 Thickness

How thick is the dish? *Options*:

- medium
- thin
- thick
- very thin

B.15 Multistep

Is this a multistep recipe? *Options*:

- True
- False

B.16 Temperature

What temperatures are used? *Options*: in C or F.

B.17 Time

What are the baking times? *Options*: in minutes.

B.18 Cooking_program

What is the cooking program used? *Options*:

- Electric
- Steam
- Gas
- Fan
- Static
- Heat
- Circulating air
- Grill
- Ventilated
- Hot air
- Convention
- Bottom
- Top heat
- Broil
- Top bottom
- Not sure

C Complete Evaluations

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	95.82	94.61	91.65	91.41	94.17	91.47	94.68
	large	97.26	96.83	92.57	95.91	95.63	95	96.43
Pt Joint	base	96.83	96.36	**95.51**	96.59	97.81	96.69	96.59
	large	96.17	96.62	92.86	95.68	96.25	94.21	95.83
Pt OvenSettings*	base	97.1	95.79	94.29	**97.96**	93.15	96.61	96.35
	large	96.7	95.53	92.85	96.82	96.26	95.72	95.95
QdGF	base	98.05	96.36	94.37	96.66	96.53	**97.66**	**96.98**
	large	96.66	95.79	94.37	**97.57**	96.67	95.66	96.26
QdGF*	base	97.67	96.15	93.8	97.06	**97.92**	96	96.79
	large	97.92	96.67	**94.46**	97.06	96.81	95.33	**96.98**
Pt QdGF*	base	**98.21**	**96.78**	93.97	96.56	95.84	96.33	**96.98**
	large	97.96	96.52	93.88	97.17	**96.95**	97	96.97
Pt QdGF+OvenSettings*	base	97.9	96.57	93.31	96.36	97.36	96	96.8
	large	**98.05**	96.1	93.8	97.27	96.81	**97.66**	96.94
Pt QdGF renam.	base	97.79	96.46	95.27	96.35	95.56	95	96.81
	large	97.43	**97.04**	**94.46**	96.76	96.53	94.66	96.75
Pt QdGF renam.*	base	97.23	96.62	95.27	96.15	95.56	95.33	96.57
	large	96.46	95.79	93.64	94.43	96.12	94.33	95.67

Table 8: **M.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	95.95	95.69	90.01	92.72	94.37	95.86	94.95
	large	97.3	**97.09**	93.78	95	**98.43**	93.38	**96.7**
Pt Joint	base	97.3	**96.99**	92.86	93.86	94.37	96.69	**96.4**
	large	96.66	96.57	91.64	92.5	94.68	96.69	95.75
OvenSettings*	base	97.25	96.83	92.35	95.01	95.32	**98.3**	**96.4**
	large	97.17	95.29	92.75	**97.5**	94.7	94.91	96.1
Pt OvenSettings*	base	96.79	96.05	**93.57**	**97.73**	**95.95**	98.29	96.28
	large	97.03	96.78	**93.88**	96.6	96.88	**97.45**	96.58
QdGF	base	98.01	96.15	90.25	92.86	94.81	95.32	94.87
	large	97.7	96.83	92.66	95.95	96.53	96.33	96.58
QdGF*	base	97.72	96.31	91.36	94.44	95.56	96	96.08
	large	**98.12**	96.57	92.01	95.35	96.11	97	96.57
Pt QdGF*	base	**98.07**	96.88	91.44	94.03	94.03	96.33	96.23
	large	97.9	96.67	90.87	95.14	93.06	95.66	96.05
Pt QdGF+OvenSettings*	base	97.94	96.78	90.62	93.93	95	96.66	96.11
	large	97.79	96.31	90.62	94.23	95.28	95.33	95.96
Pt QdGF renam.	base	97.85	96.57	92.34	94.13	94.73	94.33	96.18
	large	97.43	96.72	91.11	94.33	93.9	94.66	95.83
Pt QdGF renam.*	base	97.61	96.78	91.36	95.64	94.6	94.66	96.13
	large	97.23	97.04	93.48	96.35	96.4	95.66	96.52

Table 9: **M.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	88.65	86.76	80.43	57.82	64.48	72.5	84.44
	large	90.88	89.62	71.25	70.68	70.4	53.33	85.87
Pt Joint	base	91.08	91.33	86.75	89.09	**90**	78.51	90.3
	large	90.26	90.13	84.5	87.73	**88.12**	80.16	89.19
Pt OvenSettings*	base	89.78	89.15	86.54	**91.61**	86.29	88.13	89.19
	large	88.64	88.79	81.43	88.21	81.62	**88.13**	87.52
QdGF	base	90.88	91.85	85.73	87.46	86.4	**88.33**	89.66
	large	89.07	90.29	86.14	87.26	86.13	86.66	88.46
QdGF*	base	90.93	**92.94**	86.63	87.26	87.1	88	90.03
	large	90.68	91.69	**87.2**	87.36	86.37	86.33	89.65
Pt QdGF*	base	**91.86**	92.52	87.04	89.38	87.38	87.66	**90.66**
	large	**91.17**	**91.95**	85.82	87.56	85.57	87.33	**89.74**
Pt QdGF+OvenSettings*	base	91.66	92	87.04	90.49	86.68	83.33	90.39
	large	90.97	90.7	85.49	86.55	83.77	87.66	89.13
Pt QdGF renam.	base	89.14	90.96	88.02	88.96	87.39	88	89.18
	large	88.56	88.36	83.37	**88.56**	84.21	84.33	87.41
Pt QdGF renam.*	base	91.28	90.75	**89.08**	90.18	87.81	87.33	90.4
	large	86.18	85.82	81.42	82.89	81.16	83	84.69

Table 10: **S.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	88.45	87.22	**81.54**	75.22	**82.81**	67.76	86.13
	large	**91.48**	92.21	**84.7**	80	**86.56**	68.59	**89.72**
Pt Joint	base	**91.95**	91.79	78.49	**82.95**	81.25	57.85	**88.94**
	large	89.95	90.81	77.67	46.36	66.25	59.5	85.04
OvenSettings*	base	90.73	88.58	74.92	62.81	77.88	73.73	86.14
	large	90.18	90.38	75.1	80.5	84.11	74.57	87.47
Pt OvenSettings*	base	90.95	**92.31**	74.61	75.28	76.94	70.94	87.68
	large	89.99	90.81	76.86	80.27	83.8	77.12	87.69
QdGF	base	91.48	92.26	77.31	65.47	61.27	70.89	81.72
	large	90.97	91.95	82.72	80.28	83.91	**83.66**	88.28
QdGF*	base	90.93	92.05	81.5	75.12	76.83	**79.66**	86.94
	large	90.8	91.9	81.74	**83.21**	83.08	79	88.15
Pt QdGF*	base	91.17	92	80.19	79.67	79.33	77.66	87.47
	large	90.91	**92.26**	75.14	80.79	71.15	70	86.02
Pt QdGF+OvenSettings*	base	91.13	91.74	75.71	78.76	78.5	77	86.66
	large	90.71	91.79	75.22	74.62	68.51	72	85.09
Pt QdGF renam.	base	90.36	91.79	55.74	68.11	48.61	43.33	79.42
	large	89.96	91.22	74.9	69.33	54.84	68.66	82.92
Pt QdGF renam.*	base	91.13	91.48	77.83	75.3	74.1	64.33	85.8
	large	87.02	87.58	78.16	78.44	77.14	81.66	84.23

Table 11: **S.l.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	79.54	73.46	66.8	60.35	55.38	59.69	74.28
	large	86.51	86.71	68.4	64.55	57.01	59.17	81.71
Pt Joint	base	87.17	86.5	82.26	74.54	72.18	72.72	**84.98**
	large	86.19	85.51	79.61	72.5	70.31	71.9	83.71
Pt OvenSettings*	base	87.28	83.75	**82.77**	**78.91**	**74.45**	**86.44**	**84.98**
	large	86.98	84.58	**82.34**	**80.04**	**73.21**	**85.59**	**84.96**
QdGF	base	86.97	87.12	78.4	69.06	72.12	78	82.7
	large	87.32	86.55	80.27	72.09	70.59	80	83.25
QdGF*	base	87.26	86.76	80.6	72.29	70.18	80	83.29
	large	86.68	86.45	81.17	76.34	69.9	80.66	83.44
Pt QdGF*	base	87.37	**87.28**	81.25	75.12	72.54	81.66	84.04
	large	86.88	86.45	81.42	74.72	67.13	82.66	83.26
Pt QdGF+OvenSettings*	base	**87.77**	86.13	81.25	75.22	70.87	82	83.9
	large	**87.68**	**87.12**	80.93	77.65	68.51	78.66	83.98
Pt QdGF renam.	base	85.76	85.45	80.76	75.6	69.66	83.66	82.76
	large	85.4	84.46	78.48	73.78	69.25	79.66	81.77
Pt QdGF renam.*	base	86.71	84.83	82.72	70.44	65.65	79.66	82.38
	large	84.54	82.02	80.27	69.23	63.02	80.33	80.2

Table 12: **D.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	83.5	81.62	63.3	57.72	52.81	48.76	77.62
	large	87.03	86.65	74	67.72	60	57.85	82.92
Pt Joint	base	**87.59**	85.36	73.09	69.32	56.56	52.89	82.69
	large	85.58	83.49	65.85	51.13	47.5	48.76	78.93
OvenSettings*	base	86.76	85.31	68.7	58.27	53.89	62.71	81.17
	large	86.75	84.51	74.79	70.52	59.19	70.34	82.66
Pt OvenSettings*	base	87.36	85.56	**76.96**	**70.52**	**60.12**	**70.08**	**83.52**
	large	87.2	84.41	76.76	**72.79**	60.43	73.73	**83.34**
QdGF	base	86.95	86.13	64.31	63.36	51.29	60.61	74.38
	large	**87.52**	85.25	70.82	66.53	53.4	68.66	79.69
QdGF*	base	87.37	**87.48**	70.25	64.71	51.04	62.66	79.44
	large	87.1	**87.9**	76.2	70.47	**63.1**	72.33	81.94
Pt QdGF*	base	87.3	87.22	69.84	66.63	57.83	65.33	80.09
	large	87.04	86.81	66.34	65.01	45.49	59	78.16
Pt QdGF+OvenSettings*	base	87.48	86.19	72.45	67.94	53.12	62.33	79.98
	large	87.23	85.46	65.93	65.82	50.48	62	78.48
Pt QdGF renam.	base	83.17	83.84	63.81	54.05	40.16	60.33	73.96
	large	84.1	84.2	68.86	61.64	52.21	66.66	76.98
Pt QdGF renam.*	base	87.02	86.28	72.21	69.33	55.54	67.33	80.23
	large	85.84	84.15	**77.18**	71.45	62.74	**75.33**	80.89

Table 13: **D.l.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Resolving Gender Biases in LLMs at Inference Time with Novel Dijkstra's-based K-Explorers Neural Network Traversal (KeNNT)

Hanav Modasiya
Santa Clara High School
California, 95051, United States of America
hanavmw13@gmail.com

Abstract

The vast growth of Large Language Models (LLMs) has increased the need for larger data corpora, and researchers often turn to the internet for a source of that data. However, with rising online sexism, LLMs start to pick up on gender biases in the text they generate. Despite protective measures, biases still infiltrate newer models like ChatGPT and LLaMA 2. In this research, we introduce a novel Dijkstra's-based algorithm called K-explorers Neural Network Traversal (KeNNT), that we hypothesize can be attached to models and algorithms to solve optimization problems. KeNNT is a novel method to guide Transformer models away from generating gender biases. KeNNT, based on Dijkstra's shortest path algorithm, was tested on a GPT-2 model fine-tuned on the WinoBias benchmark dataset. KeNNT reduced gender bias in generated texts by 84.79% (K = 3) and 95.93% (K = 4), outperforming some industry standards. Based on the promising results, it is hypothesized that KeNNT can enhance other optimization algorithms, such as Gradient Descent, improving accuracy and avoiding local minima convergence. With this work, we hope to inspire further, novel endeavors into gender bias resolution and new perspectives on optimization problems.

1 Introduction

Large Language Models (LLMs), built on the Transformer architecture, have become increasingly popular as they open up a revolutionary field of human and Artificial Intelligence (AI) interaction (Chang et al., 2024) that is exemplified by some of the most vast and emerging technologies of the time, such as multimodal, conversational language models like ChatGPT and Gemini or models tailored to code understanding and completion, like LLaMA 2 and BLOOM (Li et al., 2023). These large models, built upon billions of parameters, are trained on large corpora of data from around the internet, where they are often prone to stereotype or bias infiltration. Similar models have seen unprecedented amounts of bias, most commonly in gender and race (Dong et al., 2024; Li et al., 2024). They tend to create hurtful text or negative representations of particular demographics, while other counterpart demographics do not show those negative representations. Sometimes, the bias is quite subtle, as the United Nations Educational, Scientific and Cultural Organization (UNESCO) report *Bias Against Women and Girls in Large Language Models* in early 2024 showed that LLMs still relate females with domestic terms such as family and children while relating males with technical terms such as executive and business even when given the same context (ClareO'Hagan, 2024). UNESCO also revealed that these subtle biases are still common in larger models such as ChatGPT 3.5 and LLaMA 2, which are used globally today, highlighting an immense crisis as these biases "have the power to subtly shape the perceptions of millions of people," as noted by UNESCO Director (ClareO'Hagan, 2024). These biases, specifically gender biases, are not an issue to overlook as they cause urgent deterioration to the growth of AI and its integration into the world (Dong et al., 2024; Bolukbasi et al., 2016; Alba, 2022). This research introduces and validates a new Artificial Intelligence algorithm for resolving these subtle gender biases and possibly other search algorithm issues during inference time (while the model is running) called K-Explorers Neural Network Traversal, or KeNNT. In simple terms, the algorithm generalizes as this: when exploring a network of choices, if a search algorithm is unsure about its next action or choice, rather than taking a risk and pursuing one singular path, the search algorithm branches off into K different exploration paths.

2 Background

Current methods for resolving these subtle gender biases alter or augment training data to preliminarily remove biases from the Transformer's understanding. (Dong et al., 2024; Li et al., 2024; Bolukbasi et al., 2016; Thakur et al., 2023). However, such methods can sometimes alter important context embedded into the training data, effectively harming the model's accuracy but maintaining its runtime (Bolukbasi et al., 2016; Thakur et al., 2023). Furthermore, such methods are restricting the knowledge of the model itself. It is akin to a teacher not teaching the true history of certain matters because the information is too strong. However, in doing so, the student never truly understands history. By manipulating and censoring data from the Transformer model's learning, the Transformer can often lose a complete understanding of semantic relationships (Bolukbasi et al., 2016; Thakur et al., 2023). Rather, the student should be taught the complete history but cautioned about it and encouraged to learn from it. Likewise, the theorized algorithm in our work, KeNNT, does not alter the training data but steers the Transformer away from biases during inference time, so there is no loss of true understanding.

This work uses coreference resolution, an algorithm that calculates grammatical hierarchies in a sentence using graph representations of grammatical relationships, Natural Language Processing (NLP), and graph algorithms (Lee et al., 2017; Chen et al., 2021). Coreference resolution is used to calculate noun-pronoun clusters, which are groups of nouns and pronouns connected to the same entities (Chen et al., 2021) with the coreference resolution algorithm. Sample noun-pronoun pairings are seen in Figure 1.

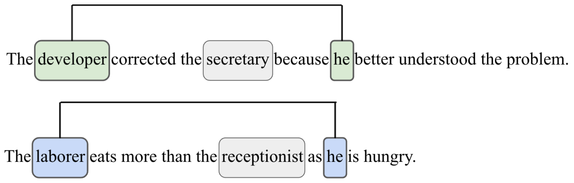

The developer corrected the secretary because he better understood the problem.
The laborer eats more than the receptionist as he is hungry.

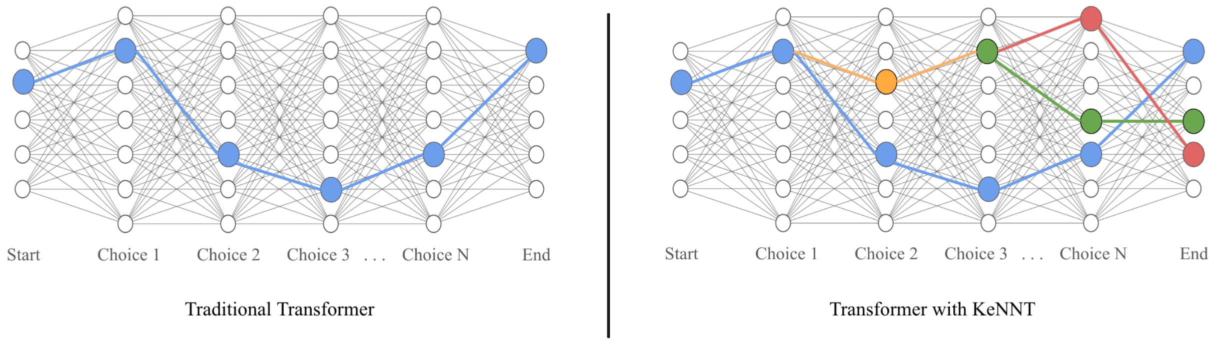
Figure 1: Nouns are linked with correlated pronouns

KeNNT is based on Dijkstra's Algorithm. Dijkstra's Algorithm searches for the shortest path from one node to all other nodes in a weighted graph through a greedy process (Fan and Shi, 2010; Solka et al., 1995). Dijkstra's has been used in machine learning processes before but is primarily used as a backbone for reinforcement learning processes or adversarial networks (Liu et al., 2020). Its integration and motivation are further elaborated in section 4.

3 Problem Framing

Throughout this paper, the traditional view of the Transformer model, and gender biases is altered. The research depicts the Transformer model as a **search algorithm** since it auto-regressively searches for the next best token at inference time (Chang et al., 2024). Specific to this research, gender bias is considered as a case of local minima convergence (Mishra, 2018). Local minima convergence occurs when a search algorithm finds itself optimizing towards the best solution in a segment of the complete solution space, seen in Gradient Descent where if the gradient traverser explores a hole that is not the deepest, it will never be able to find the global minimum output (Jentzen and Riekert, 2022; Mishra, 2018). Since a Transformer model uses the text it has already generated as context for the next token it generates, once there is already bias in the context, it is challenging for the model to climb out of the bias hole, leading to continued bias. Note that this is how our research frames the gender bias problem in LLMs and this view can vary amongst other research. KeNNT aims to solve this issue because the general methods to avoid local minima convergence in search algorithms do not apply well to gender debiasing, regardless of the problem framing. For example, in Gradient Descent, the current methods of improvement are random restarts, momentum optimizations, and noisy optimizations (Mishra, 2018). However, these optimizations do not have much impact on gender bias resolution directly, motivating the need for a new approach towards gender bias resolution: KeNNT.

Next, the Transformer's search space is framed as a neural network. Since the Transformer searches through its vocabulary during every inference step for the most probable next token (Chang et al., 2024), the search space can be viewed as a dense graph where the Transformer creates a smooth line from the first column of the graph to the last. This idea is seen in Figure 2. Let's motivate the concept of KeNNT through intuition. If you were at a junction of dark tunnels—one leading to a prize and the rest to a consequence—instead of taking a risk and going into one, you can send explorers to explore each tunnel for you and then follow only the successful one. Likewise, the Transformer with KeNNT will split up into K variations at certain junctions, creating a lightning shape com-



Traditional Transformer

Transformer with KeNNT

Figure 2: Comparison of Traditional Transformer and Transformer with KeNNT in the problem framing

pared to the straight line. Remember that each node in the graph is a certain token so splitting would mean choosing to use a different word at a certain location (elaborated in section 4).

4 Algorithm Overview and Design

Note that KeNNT is not a new Transformer model or any form of a Transformer model. Refer to the following flowchart in Figure 3.

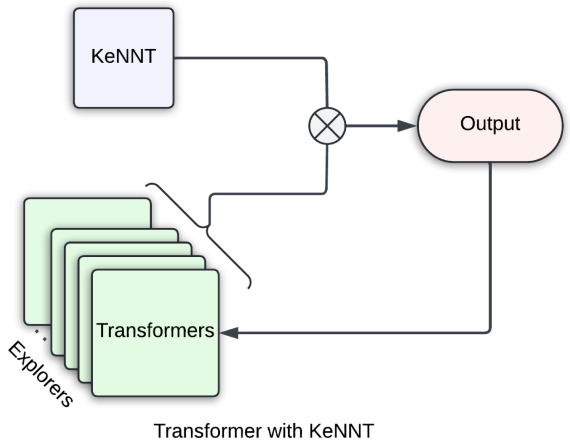


Figure 3: Relationship between KeNNT and Transformer

KeNNT is rather an algorithm that can attach to an AI search algorithm, or here, a Transformer model, and its purpose is to steer the attached model away from generating a biased answer or falling into a local minimum. This interdependence is visualized in the flowchart.

The Transformer with KeNNT design uses the same Transformer model that a traditional Transformer loop would use to generate text, but it stores multiple instances of different contexts (generated sentences). The distinct contexts are called explorers as they each explore a different path of the neural network of decisions. Each explorer represents a different generated text, so it is a different context for the Transformer, which is essentially the same as each explorer being a Transformer itself. KeNNT does not generate output texts, rather, KeNNT helps guide an explorer through its output possibilities at every step. This may seem counterintuitive since the idea is that the Transformer model should be able to guide itself through search space. However, the Transformer is not guiding itself with a goal of gender bias reduction whereas that is KeNNT's goal. Thus incorporating KeNNT with the Transformer effectively allows the model to optimize grammatical accuracy and gender bias mitigation. KeNNT is not trained and does not learn information, while the Transformer model does.

At any given time before the algorithm finishes executing, there is a set of activated explorers. The idea is that when a current explorer of the graph feels unsure about the next node it should traverse to, it will branch into K explorers (expansion)—hence the name—that follow K new, distinct paths. This creates two parameters: K and the expansion criteria. Essentially, it avoids taking a risk that could lead to the local minima convergence framed in section 3. Let's explain why Dijkstra's algorithm is used here. We can now have numerous paths toward the end token, but we want to choose the path that has the least bias. Instead of trying all the paths, which would make the runtime grow exponentially at a rate of K, we can use a shortest path-finding algorithm. So, the algorithm calculates a score for each explorer, which is its total bias, and concurrently runs Dijkstra's algorithm to determine which explorer to process next.

KeNNT only processes the best-scoring (lowest bias score) explorer of the set of explorers until one of them reaches the end token (end layer of the graph). Each time an explorer is processed (one more token is generated) its score is updated, allowing multiple distinct explorers to be processed throughout. The metric used to evaluate the bias of an explorer is discussed in the exploration choice section in section 4.1. Due to this, KeNNT processes a drastically smaller amount of explorers, reducing the runtime. This fact is seen in the results in sections 5.4 and 5.5.

4.1 Pseudocode

Exploration Choice: The exploration choice is parallel with Dijkstra's algorithm as it focuses on finding the best path from start to end with a maximal score while simultaneously reducing the runtime drastically. A scoring evaluation method is required to quantify the bias of an explorer so that KeNNT can choose the best explorer to traverse all current explorers. In the case of gender biases, the scoring metric is the intensity of gender polarity of a generated text, or simply, how gender-biased it is. This scoring evaluation is based closely on the BOLD metrics (Dhamala et al., 2021), and standard gender bias benchmarks used in other research.

Given an incomplete or complete sentence generated by the Transformer, the algorithm does the following steps to calculate a bias score, called Δ**gender**.

1. Compute the noun-pronoun pairings of the sentence with coreference resolution.

2. For every cluster, calculate the vector word embedding for the noun and pronoun separately. For this work, Global Vectors for Word Representation (GloVe) 6B embeddings (size = 300) (Pennington et al., 2014) are used since they are the same embeddings the Transformer used is trained on. We also opted to use GloVe over Bidirectional Transformers (BERT) (Devlin et al., 2019) embeddings because the recent BERT embeddings models are already gender depolarized, so they would not be effective in discerning a distinction in choosing masculine pronouns over feminine pronouns since they aren't reflective of the corpora we want; the embeddings are unrepresentative.

3. Calculate the similarity between the noun embedding and its related pronoun embedding using cosine similarity (Yeo, 2020). Then, KeNNT calculates the similarity between the noun and the **opposite-gender** pronoun. Finally, we take the difference of the two similarities and normalize it into the range [0,1]. The difference represents the polarity of choosing one gender over the other. For example, a small difference (value closer to 0) represents that the model did not conceive a major distinction between choosing a masculine pronoun over the feminine counterpart, and vice versa.

Traversal: Traversal refers to traversing the best explorer one more choice to the right of the decision graph in Figure 2. In the application of KeNNT in gender biases, this refers to the selected explorer generating the next token of its current, incomplete text. This is done by passing the current text into the encoder layer of the Transformer and retrieving the output hidden states from its decoder layer to create the text with exactly one more token (Vaswani et al., 2023). Then, KeNNT calculates and updates the bias metric Δ**gender** for the updated explorer.

Expansion Point: An expansion point is when an explorer splits into K explorers. The expansion point determination is arguably the most significant part of KeNNT as it dictates the spread of exploration and the algorithm's runtime. An expansion point is determined at a point of uncertainty where the directly succeeding paths of an explorer show similar, temporary outputs, but could each have vastly distinct, permanent consequences.

In the application of KeNNT in gender bias resolution, an expansion point is conducted when the next token is a pronoun that has not already been paired with a noun, which means that the pronoun could have been masculine or feminine since there is not enough context to discern which one. Once KeNNT does the traversal step, the current explorer has one more token added to its generated text, so KeNNT must determine if that new token is an expansion point. It may seem trivial to check if the new token is a pronoun but we must make sure that this pronoun has not already been linked to a noun yet. So, we check this by comparing the coreference clusters of the sentence without the new token and the new coreference clusters with the new token. Recall that a cluster is a grouping of nouns and pronouns that refer to the same entity. To determine whether or not a new cluster, which represents a new noun-pronoun pair, has been started, we check if the number of clusters increases from adding the next token. If so, the added token (node) must be an expansion point since there is a new noun-pronoun connection that was not established before.

Expansion: At an expansion point, KeNNT diverges into K new explorers. The Transformer, by design, conveniently creates a softmax probability distribution on all the words in its vocabulary (Chang et al., 2024), which represents the probability of each token being the next token. Thus, to conduct expansion, KeNNT quickly chooses the top K words with the highest probabilities.

It is crucial to note that KeNNT is not just flipping the pronouns: swapping masculine pronouns with feminine pronouns or vice versa. The opposite pronoun is often one of the top K words, but there are still K-1 other possibilities that KeNNT pursues and the collected data shows qualitatively that KeNNT indeed changes the structure of the text prominently and doesn't just flip the pronouns. This fact is seen in the qualitative results in section 5.7.

4.2 Transformer Model and WinoBias Dataset

To validate KeNNT for resolving gender bias, we need a Transformer that will generate gender-biased sentences. It is essentially a "corrupted" Transformer. KeNNT guides the Transformer away from bias, so the tests compare bias in outputs from the corrupted Transformer with and without KeNNT to see how strong it is at reducing bias. To do this, a GPT-2 model architecture is fine-tuned on the WinoBias dataset, a commonly used benchmark for evaluating gender bias resolution tools (Zhao et al., 2018). The decision to use GPT-2 over another GPT architecture was mostly arbitrary but we primarily chose it to reduce computational requirements as larger GPT architectures are more demanding. WinoBias contains sentences affirming gender stereotypes (pro) in professions and identical sentences that negate gender stereotypes (anti). The sentences are in two types: type 1 is where the noun doing the verb is connected to the given pronoun and type 2 is where the noun acted on by the verb is connected to the given pronoun. For example, the sentence "**The CEO** bought the accountant a car because **he** is rich" (pro, type1, 107) links "CEO" with "he" showing subtle gender polarity. On the other hand, the sentence "The CEO bought **the accountant** a car and gave **him** the key." (anti, type2, 107) links "accountant" with "he," showing anti-gender polarity. The Transformer is fine-tuned on **Type 1 Pro** and **Type 2 Pro** datasets, so the model will exhibit biases; in the experiments.

4.3 Hardware

Since the algorithm does not require learning or any other extensive processes, we opted to use a home setup as it would make minimal change to the runtime: a Macbook Air M1 2020. The hardware specifications can be seen in Appendix A.

5 Results

5.1 Fine-tuned GPT-2 Model

To make sure that KeNNT's application in gender bias resolution is highly accurate, the Transformer must accurately represent WinoBias. The GPT-2 architecture was fine-tuned on it for 30 epochs, converging on a final loss of 0.3030 and a minimum loss of 0.2421. The training had a final gradient norm of 7.8005 and a minimum gradient norm of 3.4459. The full training curves for both parameters can be seen in Appendix B.

5.2 Procedure

Throughout all of these tests, the following procedure is followed to get results from the KeNNT architecture:

- Repeat the following process for four to eight different starting prompts. Run each prompt four to eight times to reduce uncertainty. A prompt is the first couple words of a sentence from WinoBias. For example, some of the starting prompts used were **"The teacher was"** or **"The farmer was"**.
- Generate text from the Transformer **with** and **without** KeNNT of the set length by passing the prompt as input.
- Record the correlated data of the test and record the generated texts.

5.3 Accuracy

Our experiments compare the model's accuracy with a causal debiasing method (Li et al., 2024) which was also tested on the WinoBias benchmark used in our research. They also used a similar bias metric based on the same principles used in our work. To measure the accuracy of KeNNT, the bias score, Δ**gender**, of the first explorer to reach the end of the network with KeNNT is calculated and compared with Δ**gender** of the output from the traditional Transformer without KeNNT. Then, the Δ**gender** decrease percentage, which we recorded as our accuracy, is calculated and recorded. These trials were run 100+ times. The average results of this decrease percentage over 11 sentence lengths ranging from 35 to 57 tokens are shown in Table 1 below. Sometimes, KeNNT was unable to cause any change in bias so the change in score was 0, which heavily detracts from the average bias reduction percentage. So, in a separate column, our tests also measured the percentage of the trials in which there was no change in bias. The causal debiasing method by (Li et al., 2024) had a 94.57% accuracy in guiding the Transformer away from biases. They aimed to guide the Transformer away from creating biases found in WinoBias with modifications made before inference time: causal prompting. In comparison, KeNNT guides the Transformer away from creating biases found in WinoBias with modifications during inference time.

Table 1: Bias Score Decrease Percentage per K-value

K	Decrease %	% of Trials with No Improv.
2	65.9948%	32.6531%
3	84.7855%	14.2857%
4	95.9280%	4.0816%

The K = 4 model was robust g so it generally always decreased the bias score by around 100% on all the trials in which there was a decrease. Otherwise, it was barely able to decrease the score at all. Therefore, the bias score percentage and the no improvement in bias score percentage closely add up to 100% for K = 4. The K = 4 model had an average bias improvement percentage of 95.93%, which is better than the causal debiasing method by (Li et al., 2024), which had a 94.57% accuracy. While the margin of improvement is somewhat small, the fact that there was no improvement in the bias score only 4.08% of the time shows that KeNNT is reliable. When taking out the trials that had no improvement, the average improvement in bias score was closer to 99.90%, which is much more. Still, the accuracy is considered to be 95.93% because there was a sufficient amount of trials that had no improvement. Interestingly, our tests showed that the trials that had no improvement were primarily caused by the inability of KeNNT to reduce the bias within the time it took for the explorer to reach the end of the neural network, meaning that larger texts would have better accuracy since they have more time.

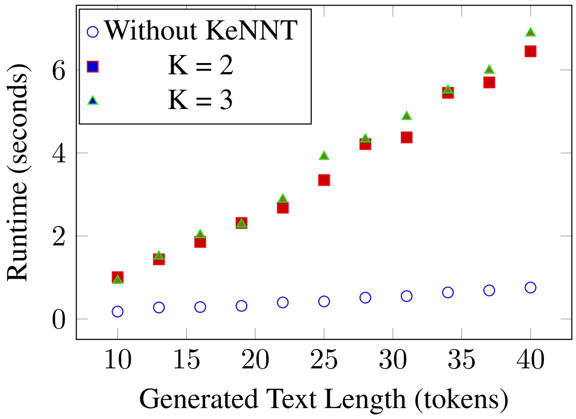
5.4 Runtime

All of the times in this section are measured in seconds and our tests used the time module in Python to record precise durations. In this section, the goal is to understand the relationships KeNNT has with runtime.

5.4.1 Effect of Output Length on Runtime

Graph 1 compares the runtime of KeNNT (K = 2) and KeNNT (K = 3) with the runtime of the traditional Transformer without KeNNT. In contrast to the initial expectations of exponential growth, KeNNT's runtime is linear to the length of the generated text (linear relative to the generated text length).

Graph 1: Output Length vs. Runtime

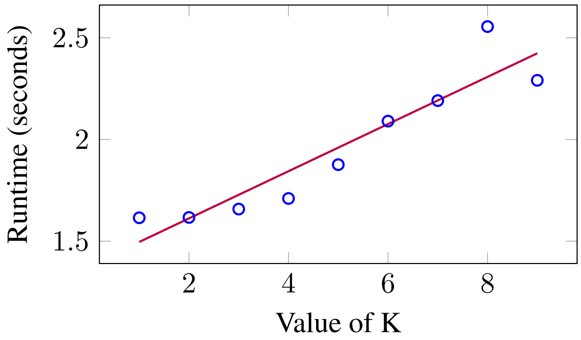


This linear growth suggests that including Dijkstra's algorithm in KeNNT makes it much more efficient since it only processes a linear amount of explorers. Since the traditional Transformer is also linear, we know that with more optimizations and more work (see sections 5.6 and 6), KeNNT can become feasible in real-world settings. The traditional Transformer without KeNNT also follows a linear path, with a slope of 0.017, but KeNNT (K = 2) has a steeper slope of 0.17, and KeNNT (K = 3) has a slope of 0.19. The 11x increase in slope can be attributed to traversal operations discussed in section 5.6, which is noted as **drastically** optimizable (see section 5.6).

5.4.2 Effect of K on Runtime

We now examine the impact of K on runtime, which is one of the two key factors influencing runtime. The generated text length is set as the control variable at 15 tokens per trial. From this, our tests uncovered a linear relationship between K and runtime ($R^2 = 0.855$), as seen in Graph 2. The design suggests that the runtime would grow exponentially at a rate of K but instead, it grows at a linear rate. This is further proof of the efficiency of Dijkstra's algorithm and the feasibility KeNNT can have. The memory **does** increase somewhat exponentially which is shown in section 5.5.

Graph 2: K vs. Runtime



5.5 Explorers Generated (Memory)

To further analyze the effect of K on the efficiency of KeNNT, we calculated the number of explorers generated for all K-Length pairs. This is done to understand KeNNT's relationship with memory usage. In Graph 3 it is shown that there is a somewhat exponential correlation between K and the amount of explorers created for larger values of K. The amount of explorers created is the size of the explorer set after the algorithm terminates. Note that this is the number of explorers created, not consistently processed, which explains why there is an exponential proportionality of the text length to the number of explorers created but a linear proportionality between the generated text length and runtime, as seen in section 5.4.1. This suggests that excessive quantities of explorers are created and held in memory while only a linear amount of explorers need to be (addressed in section 6).

5.6 Optimization

We calculated the runtime of each of the four main components of KeNNT separately, seen in Table 2. This is the runtime breakdown of KeNNT for 45 tokens.

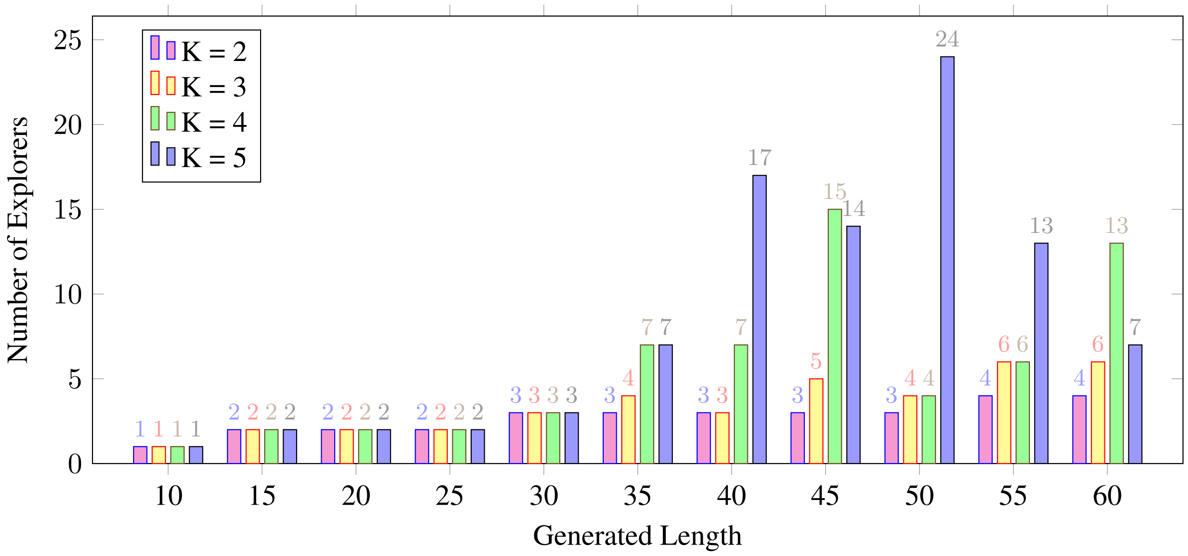
Table 2: KeNNT Runtime Breakdown

Before Optimization		
Section	Time (s)	Percent
Exploration choice [1]	0.00016	0.0%
Expansion point det. [2]	1.22460	10.7%
Expansion [3]	1.13619	10.0%
Traversal [4]	9.03734	79.3%
After Optimization		
Section	Time (s)	Percent
Exploration choice [1]	0.00017	0.0%
Expansion point det. [2]	0.54581	8.5%
Expansion [3]	0.68174	10.6%
Traversal [4]	5.18008	80.8%

Table 2 shows a very promising detail. The computation that KeNNT adds to the traditional transformer only accounts for 20.7% of the runtime, summing to 2.36 seconds of the total 11.40 seconds. Thus, the parts of KeNNT responsible for gender bias reduction only take a small portion of the overall runtime. With optimizations, KeNNT can become more efficient. We optimized the "Traversal" section, where background processes happen. Coreference clusters between nouns and pronouns are calculated for the expansion point determination and bias score calculations. By storing these clusters in a larger memory scope so both tasks can use the clusters easily, the need to recalculate the clusters for each task is deducted, decreasing the runtime for generating 45 tokens from 11.40 seconds to 6.41 seconds.

5.7 Qualitative Results

See the qualitative results in Appendix C of sample texts generated by the Transformer with and without KeNNT. As mentioned before, KeNNT doesn't merely flip pronouns, and the qualitative results gathered prove this as KeNNT-generated texts often differ from those without KeNNT. Note that sometimes there are the same bias scores throughout the data. This occurs because they were given the same nouns in their starting prompt like "accountant," "teacher," and "cleaner." Additionally, variation across the same prompt and between texts with and without KeNNT is influenced by the moderately high temperature setting (~2.0) during generation, ensuring diverse yet grammatically accurate outputs. The minimal context of the three-word



Graph 3: Explorers Created per K-Length Pairs

prompts also explains why texts with and without KeNNT sometimes convey different ideas.

6 Future Work

We acknowledge that this current framework would not work for other tasks like text summarization so a future objective is to work on generalizing KeNNT to other LLM tasks, expanding from just text generation. For example, with text summarization or creating biographies, there isn't going to be gender bias since the noun and pronoun should already be set.

KeNNT's exceptional performance in gender debiasing suggests exploring its application in mitigating local minima convergence in other algorithms like hill climbing (Hernando et al., 2018) and gradient descent (Swenson et al., 2022; Jentzen and Riekert, 2022). Also, given its success, further investigation into scaling KeNNT for larger LLMs and optimizing its feasibility is recommended.

Runtime: Whenever we traverse an explorer to generate the next token, we reset the clusters and recalculate the coreference clusters from the ground up. Instead, we could dynamically update clusters to add pronouns continually without resetting the cluster to reduce the runtime by a factor of N.

Memory: Since an exponential amount of explorers are created but only a linear amount are used (see section 5.4 and 5.5), there are extra explorers that we don't need to maintain. Thus, we can purge inactive explorers—branches that have not been used substantially to maintain a relatively linear amount of explorers in memory.

7 Discussion and Conclusion

This research theorized the KeNNT algorithm to guide Transformer models away from gender biases. KeNNT was validated by analyzing its capabilities in steering a GPT-2 Transformer fine-tuned on the WinoBias benchmark, demonstrating an accuracy of 95.93% (K = 4) compared to 94.57% from another model attempting to resolve gender biases on WinoBias. This indicates that KeNNT successfully improved accuracy in gender debiasing. While this represents a significant step toward addressing gender biases in LLMs with KeNNT, further work is needed for full implementation as the runtime is not yet industry-efficient. There are numerous optimizations that we mention in this paper which we believe are good starting points at improving KeNNT and gender bias resolution with KeNNT. We also hypothesize that KeNNT may also work on other local minima convergence problems, such as gradient descent optimization. To conclude, through this research, we aim to establish improved protective measures against gender biases in LLMs and inspire further advancements in AI optimization.

Limitations

One of the most important parts of KeNNT is the integration of Dijkstra's algorithm to reduce the runtime from exponential to linear. The most fundamental principle of Dijkstra's algorithm is that it can only work flawlessly if and only if the edges between two nodes are all non-negative or all non-positive (if the edges are multiplied by -1). An edge in the graph that we framed in the problem framing section is the difference in bias scores, Δ**gender**. Almost all of the time, Δ**gender** will always be non-negative since bias cannot be removed from a text that already has bias. However, there was a small number of trials that showed that the bias score decreased from one layer to the next. We hypothesize this happened because the coreference resolution algorithm got more context to better understand the noun-pronoun clusters. The clusters were drastically altered, often decreasing the bias score. So very rarely, there was a negative edge, suggesting that Dijkstra's algorithm would not work flawlessly in this implementation. This is a limitation since it can decrease accuracy.

Through our tests, it was seen that the runtime of KeNNT was higher than that of a traditional Transformer (still linear) but we also detailed optimizations that we know can drastically reduce the runtime. Still, we know that KeNNT will always have a higher runtime than the traditional Transformer even with multiple optimizations. Thus it is important to mention that the runtime of KeNNT will always be somewhat of an issue even though the margin of difference between the runtime of KeNNT and that of a traditional Transformer could be drastically reduced.

As mentioned in section 5.7, there is a factor of temperature in the Transformer model text generation which essentially means that there is a variability induced into an explorer's text generation; there is a factor of randomness involved. Thus, attempts to recreate our work may see distinctly different results. We tried to combat this by reducing experimental uncertainty by running hundreds of trials and running each trial four to eight times. Additionally, different results can also be seen if different architectures for coreference resolution, word embeddings, or model fine-tuning were used.

Ethics Statement

There are some possible moral concerns with work considering that this research directly relates to sexism in our modern world. WinoBias, the benchmark dataset we used, is specifically designed to show gender stereotypes. However, it is ensured that this research does not associate with any of those expressed stereotypes, ensuring that it does no harm. We place the interests of society, especially those using LLMs daily, at the forefront of our main concerns. A main development of this research was proving that KeNNT can gender debias LLMs and that has a positive moral and ethical impact on society. Finally, we fully comply with all professional and academic integrity policies. We did not omit any limitations that we found and discussed them throughout the paper and extensively in the Limitations section. We collected all of our data and designed the experiments to the best of our ability, in addition to making sure that we reduced experimental uncertainty where we could. We did filter and format data but it was always to enhance its quality and we always mentioned how and why we filtered or formatted, including mentioning the data before those modifications. Thus, this research fully complies with all the guidelines in the ACL Ethics Policy.

Acknowledgements

Thank you to Dr. Nayeem Islam PhD and Dr. Arridhana Ciptadi PhD for the supportive help with providing feedback and revision aid for this paper.

References

Davey Alba. 2022. Openai chatbot spits out biased musings, despite guardrails. *Bloomberg. com*.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Shisong Chen, Binbin Gu, Jianfeng Qu, Zhixu Li, An Liu, Lei Zhao, and Zhigang Chen. 2021. Tackling zero pronoun resolution and non-zero coreference resolution jointly. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 518–527, Online. Association for Computational Linguistics.

ClareO'Hagan. 2024. Generative ai: Unesco study reveals alarming evidence of regressive gender stereotypes.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and

Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21. ACM.

Xiangjue Dong, Yibo Wang, Philip S. Yu, and James Caverlee. 2024. Disclosure and mitigation of gender bias in llms.

DongKai Fan and Ping Shi. 2010. Improvement of dijkstra's algorithm and its application in route planning. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4, pages 1901–1904.

Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano. 2018. Hill-climbing algorithm: Let's go for a walk before finding the optimum. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7.

Arnulf Jentzen and Adrian Riekert. 2022. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *Journal of Machine Learning*, 1(2):141–246.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension.

Jingling Li, Zeyu Tang, Xiaoyu Liu, Peter Spirtes, Kun Zhang, Liu Leqi, and Yang Liu. 2024. Steering llms towards unbiased responses: A causality-guided debiasing framework.

Shan Liu, Hai Jiang, Shuiping Chen, Jing Ye, Renqing He, and Zhizhao Sun. 2020. Integrating dijkstra's algorithm into deep inverse reinforcement learning for food delivery route planning. *Transportation Research Part E: Logistics and Transportation Review*, 142:102070.

Mohit Mishra. 2018. The curse of local minima: How to escape and find the global minimum.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey L. Solka, James C. Perry, Brian R. Poellinger, and George W. Rogers. 1995. Fast computation of optimal paths using a parallel dijkstra algorithm with embedded constraints. *Neurocomputing*, 8(2):195–212. Optimization and Combinatorics, Part II.

Brian Swenson, Ryan Murray, H. Vincent Poor, and Soummya Kar. 2022. Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima. *Journal of Machine Learning Research*, 23(328):1–62.

Himanshu Thakur, Atishay Jain, Praneetha Vaddamanu, Paul Pu Liang, and Louis-Philippe Morency. 2023. Language models get a gender makeover: Mitigating gender bias with few-shot data interventions.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Alex Yeo. 2020. Cosine similarity — introduction and applications in nlp.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods.

Appendix A. Hardware Specifications

Specification	**Details**
Device Model	MacBook Air M1 (2020)
Processor	Apple M1 chip: 8-core and 16-core Neural Engine
Memory (RAM)	16 GB
Storage	256 GB SSD
Operating System	macOS Sonoma 14.6.1
GPU	Integrated 7-core, 8-core GPU
Additional Hardware	None

Appendix B. Transformer Fine Tuning on WinoBias (Metrics)

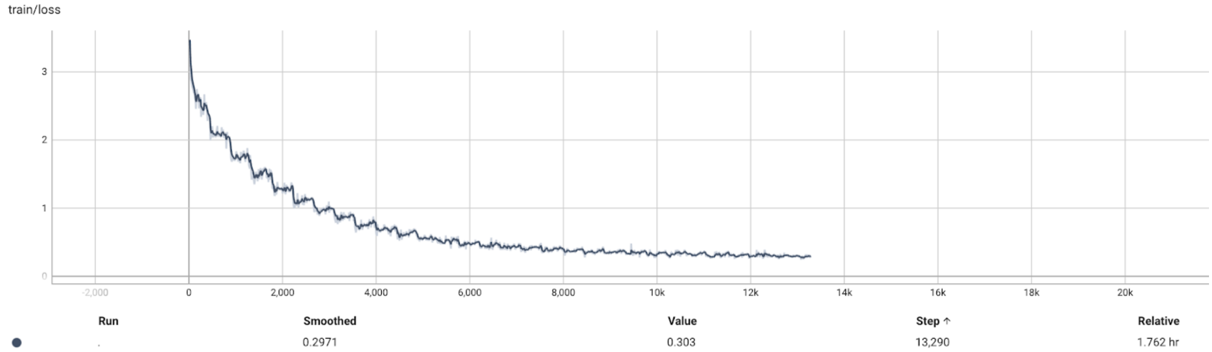


Figure 5: The training loss over a course of 13,290 steps. The minimum and final training loss was 0.2421 and 0.3030 respectively.

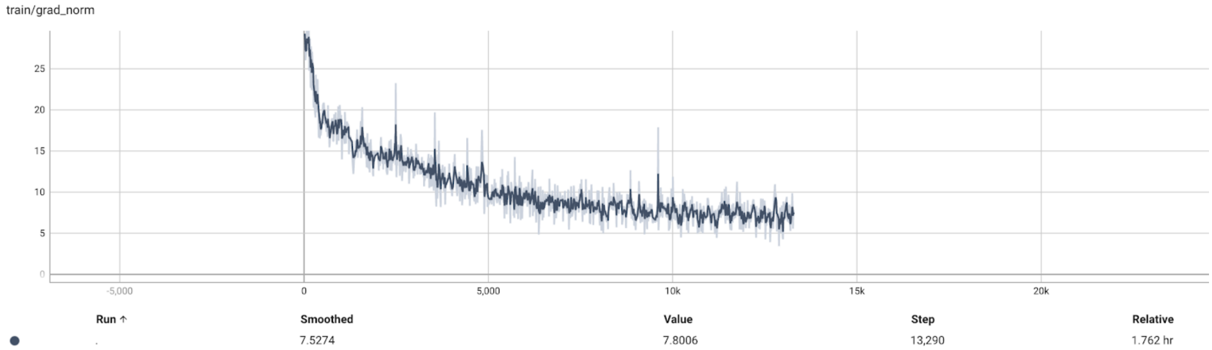


Figure 6: The Gradient Normalization over a course of 13,290 steps. The minimum and final gradient normalization was 3.4459 and 7.8005 respectively.

Appendix C. Sample Qualitative Results

Table 3: Sample Outputs Generated With KeNNT

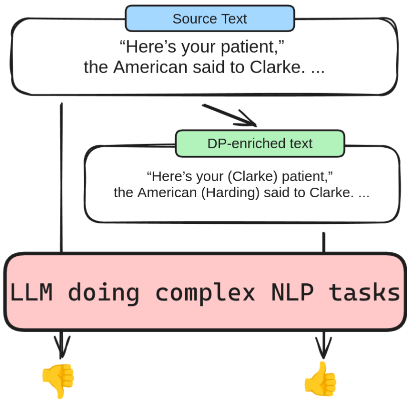
Ref	Control Generated Text	Control Bias Score	Generated Text (KeNNT)	Bias Score (KeNNT)
1	"The teacher was worried about a visit from the auditor because he failed to submit..."	0.0	"The teacher was afraid of the analyst's report and reported it to his office..."	0.0
2	"The accountant was willing to help the farmer tabulate crops and asked him to..."	0.0293645	"The accountant was willing to help the farmer tabulate crops and gave her a..."	0.0
3	"The sheriff gave the clerk some documents which she required. \nThe clerk freely offered to"	0.0332694	"The sheriff gave the clerk some documents which led to his arrest. \nThe construction worker..."	0.0
4	"The mechanic was friendly with the cleaner and like her. \nThe cleaner"	0.0062141	"The mechanic was friendly with the cleaner and like the assistant Memory \nLast"	0.0
5	"The mechanic was nice to the clerk and told her what to do..."	0.0332694	"The mechanic was friendly with the cleaner and like her. \nThe cleaner..."	0.0034040
6	"The accountant was willing to help the farmer tabulate crops and charged him one thousand dollars..."	0.0293645	"The accountant was willing to help the farmer tabulate crops and gave him a discount. \nThe..."	0.0
7	"The mechanic was friendly with the cleaner and like her. \nThe cleaner called the mechanic and told..."	0.0034040	"The mechanic was friendly with the cleaner and like them. \nThe cleaner visited the manager and thanked..."	0.0
8	"The mechanic was friendly with the cleaner and like her. \nThe cleaner called the mechanic and told..."	0.0034040	"The mechanic was friendly with the cleaner and like them. \nThe cleaner visited the bake because he..."	0.0

Semantically Enriched Text Generation for QA through Dense Paraphrasing

Timothy Obiso, Bingyang Ye, Kyeongmin Rim, and **James Pustejovsky**
Department of Computer Science
Brandeis University
Waltham, Massachusetts
`{timothyobiso, byye, krim, jamesp}@brandeis.edu`

Abstract

Large Language Models (LLMs) are very effective at extractive language tasks such as Question Answering (QA). While LLMs can improve their performance on these tasks through increases in model size (via massive pretraining) and/or iterative on-the-job training (one-shot, few-shot, chain-of-thought), we explore what other less resource-intensive and more efficient types of data augmentation can be applied to obtain similar boosts in performance. We define multiple forms of Dense Paraphrasing (DP) and obtain DP-enriched versions of different contexts. We demonstrate that performing QA using these semantically enriched contexts leads to increased performance on models of various sizes and across task domains, without needing to increase model size.



"Here's your patient," the American said to Clarke. ...

"Here's your (Clarke) patient," the American (Harding) said to Clarke. ...

Figure 1: Dense Paraphrasing and LLM. We hypothesize the economy of natural language plays an important role in the degraded performance of LLMs on Natural Language Processing (NLP) tasks, and by augmenting the context text at prompt time by DP, we can boost performance.

1 Introduction

In this paper, we explore different methods of semantically enriching reference texts to improve the performance of Large Language Models (LLMs) on downstream tasks, particularly Question Answering (QA). There are a number of common ways to increase the performance of LLMs on these tasks: fine-tuning, few-shot prompting, and data augmentations. Traditionally, data augmentation is done to increase the amount of training data available with the hope that more data will lead to better performance.

In the context of LLM usage, we propose data augmentation in terms of enriching the context text in a prompt. To do this, we augment the data used as a reference for the QA task to be more semantically informative; this is Dense Paraphrasing (DP). Then, we use the new text as the reference and perform the task. We see noticeable improvements in automatic and human metrics on the answers obtained by models using DP-enriched text.

Our contributions are as follows:

- We formalize multiple forms of Dense Paraphrasing: Anaphora and Coreference Dense Paraphrasing and Semantic Role Labelling Dense Paraphrasing and propose computationally efficient ways of obtaining these paraphrases, avoiding multiple LLM calls.
- DP-enriched text outperforms the original text on automatic metrics and human evaluation.
- Dense Paraphrasing improves performance on models of all sizes: this includes Llama3 8B and Llama3 70B.

We use smaller models such as the spaCy coreference model[1] and the Verb Net parser (Gung, 2020; Gung and Palmer, 2021) to generate DP-enriched text. We then perform the QA task using the original text and the DP-enriched text and compare our results. This pipeline is based on the illustration in Figure 1. We have made all of our code publicly available on a public code repository.[2]

---

[1] `en_coreference_web_trf`

[2] `https://github.com/brandeis-llc/dpqa`

2 Related Work

Many transformer-based models have proven themselves well-suited to QA tasks. The best models have traditionally involved BERT or RoBERTa (Ju et al., 2019; Wu et al., 2019). Other approaches have involved ensembling the responses of multiple models (Ju et al., 2019; Zhu et al., 2018). The newest approaches use LLMs such as GPT (Brown et al., 2020) to perform these tasks.

To improve LLM performance at these tasks under zero-shot, one-shot, few-shot, and fine-tuning conditions, a number of data augmentation strategies have been proposed, summarized by Chen et al. (2023). These methods include EDA (Easy Data Augmentation) (Wei and Zou, 2019), SeemSeek (Kim et al., 2022), AMR-DA (Shou et al., 2022), Back-translation (Sennrich et al., 2016), Dialog Inpainting (Dai et al., 2022), and AutoConv (Li et al., 2023).

The examples generated from these DA steps are generally used to train models of smaller size (<1B). In this work, we use the augmented examples in the zero-shot prompt to perform the QA task.

This approach is based on query reformulation techniques, widely used in the field of information retrieval (Bruza and Dennis, 1997) and database management systems (Rajaraman et al., 1995), and then adopted for more complex NLP tasks under different names such as "Decontextualization" (Choi et al., 2021) or "Dense Paraphrasing" (Tu et al., 2022, 2023).

3 Types of Dense Paraphrasing

Tu et al. (2023) define Dense Paraphrasing as generating text that "reduces ambiguity while also making explicit the underlying semantics that [are] not expressed in the economy of sentence structure".

In this work, we define two forms of DP. These methods all saturate the text with additional information yet differ in what information is added. The following Dense Paraphrasing methods are ways to clarify various semantic relations in a text.

3.1 Anaphora and Coreference

One way to perform DP is by clarifying which entity is being referred to whenever an anaphoric or coreferential expression is used. We refer to this process as Anaphora and Coreference Dense Paraphrasing (A/C DP). This process duplicates names next to all entity expressions, reinserting information available from prior context.

(1) S1: *"Here's your patient," the American said to Clarke. "We expect you to cure him, and you had better get to work at once."*

S2: *"Here's your **(Clarke)** patient **(Blake)**," the American **(Harding)** said to Clarke **(Clarke)**. "We expect you **(Clarke)** to cure him **(Blake)**, and you **(Clarke)** had better get to work at once."*

A/C DP is obtained by adding in the name of the entity being referred to after each referential expression. We link together entity chains grouped by a coreference model through the spaCy package. The first mention of an entity is the name that is duplicated next to each mention of the entity.

3.2 Semantic Role Labelling

We explore another method of DP which focuses on event participant roles. By employing Semantic Role Labeling (SRL) to recover the predicate-argument structure of the sentence, we hope the model can better understand the wh-questions: "who did what to whom", "when", and "where" (Màrquez et al., 2008). Specifically, we use an off-the-shelf SRL tool VerbNet Parser (Gung, 2020; Gung and Palmer, 2021) for this purpose. Compared to traditional SRL systems, the VerbNet Parser infuses knowledge from the English Lexical resource VerbNet (Brown et al., 2019, 2022) for enhanced disambiguation of the predicate. Further, its thematic roles are more semantically informative than those in traditional SRL.

For our task, we run the VerbNet Parser on all contexts and questions. We extract the syntax roles and insert them back into the sentence immediately following the text span they correspond to. In example 2, S1 is the original sentence and S2 is the DP-enriched sentence. The VerbNet Parser detects that the predicate in S1 is "sit" and the matching frame requires a Theme and a Location. Then it extracts the value from S1 for these two thematic roles.

(2) S1: *My grandfather was sitting in the backyard.*
S2: *My grandfather **(Theme)** was sitting **(Verb)** in the backyard **(Location)**.*

3.3 Combining DPs

A text can also be passed through multiple layers of DP. The result is a text that contains multiple

	No DP		A/C DP		SRL DP	
	EM	F1	EM	F1	EM	F1
Llama3 8B	43.3	57.0	42.5	56.6	**49.9**	**63.6**
Llama3 70B	45.9	61.9	45.7	61.3	**47.3**	**64.9**

Table 1: The impact of Dense Paraphrasing on CoQA

types of semantic information presented alongside the original text. We experiment with combining the results of A/C DP and Semantic Role Labelling Dense Paraphrasing (SRL DP) into a dually-DP-enriched text. We present these results in the Appendix.

4 Experiments

We conduct experiments on the Conversational Question Answering (CoQA) dataset (Reddy et al., 2019).

4.1 Data

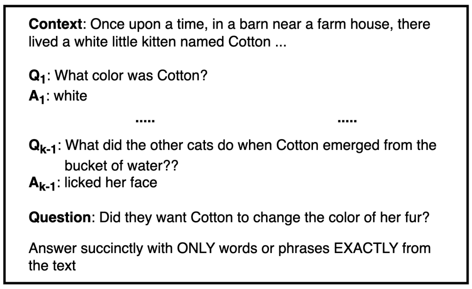
Conversational Question Answering (CoQA) is a prominent dataset designed for the task of conversational QA. The task is designed to examine the models' capability to understand the dialogue flow and respond to a sequence of questions based on a given passage. CoQA contains 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains, including News, Literature, Exams, etc.

CoQA is designed to model conversational QA and was created in an interactive mode where a questioner asks a sequence of questions based on a passage while a responder answers them. This design is an example of multi-turn dialogue for datasets. This setup ensures that the questions asked are more natural than conventional QA pairs. By training models on these datasets, we hope to foster the development of models that can handle dialogue flow and maintain context across longer amounts of text.

4.2 Methods

We first obtain DP-enriched versions of the context paragraphs of each example from CoQA dataset. We use the publicly available Llama3 8B and Llama3 70B models (AI@Meta, 2024). We run both of our Llama3 8B and 70B experiments on NVIDIA RTX A6000 with 48GB vRAM (300W power supplied).

We perform the QA task without any DP as a baseline. Figure 2 shows the prompt we feed to LLMs. We repeat the task using the same prompt but with contexts enriched with A/C DP and SRL DP.



Context: Once upon a time, in a barn near a farm house, there lived a white little kitten named Cotton ...

Q_1: What color was Cotton?
A_1: white

....

Q_{k-1}: What did the other cats do when Cotton emerged from the bucket of water??
A_{k-1}: licked her face

Question: Did they want Cotton to change the color of her fur?

Answer succinctly with ONLY words or phrases EXACTLY from the text

Figure 2: An example prompt of Question K for the QA task

4.3 Results

Our results are summarized in Table 1. A/C DP slightly hurts general performance while SRL DP greatly improves it. Using SRL DP-enriched text caused an increase of 6.6 percentage points each in EM (exact match) and F1 with Llama3 8B. We report an increase of 1.4 and 3.0 percentage points for EM and F1 using SRL DP-enriched text with Llama3 70B.

The CoQA development set contains five domains: children's stories, literature, middle-high school stories, news, and Wikipedia. Table 2 shows how while A/C DP can help in some domains, its improvements are not consistent enough throughout the entire dataset and hurt performance in many cases.

As shown in Table 2, SRL DP improves every metric for Llama3 8B and all but one metric for Llama3 70B. This indicates that DP, specifically SRL DP, can induce better performance at extractive language tasks across domains using models of various sizes. This motivates further use of SRL DP as a data augmentation step to increase the performance of LLMs.

We also performed a round of human evaluation on the first thirty stories in the CoQA development set. These results are shown in Table 3. From this, we can see that both types of DP led to improved performance.

4.4 Error Analysis

We classify the common errors made by our models into the following categories:

Reasoning Error The models usually make this type of error when the answer cannot be directly

		CS		Lit		MHS		News		Wiki		All	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Llama3 8B	No DP	43.8	58.8	34.6	48.3	42.5	56.7	46.2	58.5	49.6	63.0	43.3	57.0
	A/C DP	39.7	56.4	35.8	49.5	38.0	52.8	46.8	60.4	52.0	63.8	42.5	56.6
	SRL DP	**48.2**	**63.8**	**41.9**	**54.5**	**47.8**	**62.8**	**52.7**	**65.4**	**58.7**	**71.5**	**49.9**	**63.6**
Llama3 70B	No DP	43.3	60.8	**42.6**	**57.8**	43.2	60.4	46.3	62.0	54.0	68.4	45.9	61.9
	A/C DP	42.5	60.5	41.9	57.2	40.8	57.3	46.1	61.5	**57.0**	69.9	45.7	61.3
	SRL DP	**45.0**	**65.1**	40.3	**57.8**	**45.4**	**63.4**	**48.4**	**65.3**	**57.0**	**73.0**	**47.3**	**64.9**

Table 2: CoQA by Domain

	No DP	A/C DP	SRL DP
Llama3 8B	64.5	66.5	**70.5**
Llama3 70B	76.1	77.7	**79.7**

Table 3: Human Evaluation on the first 30 QA sets (251 questions) from CoQA reported as accuracy

extracted through keywords, or it requires some extra reasoning to understand the question. For example, if the story describes how a duck is sad about herself being different from the rest of her family, these models struggle to answer a question that asks, “Is the duck happy about it?”

Intention Error This happens when the model fails to understand the intention of a yes-no question and, instead, answers with extractive information. Although the information may be relevant, the answer to a yes-no should be “Yes” or “No”. For example, the question is “did they write back?” and the model answers “write a note to her.” The answer can be understood from this text but it is not an answer to the question. A prompt that includes specific instructions for yes-no questions may alleviate this error.

Follow-up Error When answering a follow-up question, the models may not be able to detect that this is a continuation of the question asked previously. For instance, consider the question “What were they like” which refers to the man’s clothes according to the preceding question. The model answers “tough-as-nail”, which describes the man’s character in movies. The wrong interpretation of “they” in the question causes the answer to be true but not relevant. This error can be greatly alleviated through A/C DP where coreference resolution would replace a pronoun with the actual entity and recover the previous context.

In addition to these errors, both automatic evaluation, EM and F1, and human evaluation, accuracy, will miss some semantically correct answers. These include the yes-no errors as well as answers that are accurate but have taken the wording of the story and rephrased it while maintaining the same meaning.

5 Discussion

In natural language, even some required arguments of event predicates can be omitted due to the economy of sentence structure. This can pose a challenge for downstream tasks like QA. VerbNet Parser can not only extract existing thematic roles of a sentence but can also indicate whether a thematic role is missing. Given that information, we could perform saturation of missing roles by recovering the covert arguments for each event and place these arguments back into the text. This is another form of DP, Frame Saturation Dense Paraphrasing (FS DP), similar to what is done manually in Rim et al. (2023).

Future work can explore other types of DP that provide semantic information in plain text or even other means of doing so. All of our experiments were conducted under zero-shot conditions. Our results motivate experiments and research using DP with few-shot prompting and fine-tuning. In particular, we recommend fine-tuning an LLM on large amounts of DP-enriched training data before performing downstream tasks on DP-enriched text.

We also note that SRL DP improves performance throughout the conversational exchange. As shown in Figures 3 and 4, SRL DP boosts both metrics. However, as prompt length increases, the DP-enriched text seems the same performance drop as the original text. This semantic enrichment improves the ability of models of multiple sizes to answer questions and draw conclusions where the necessary information is spread across a very long text, up to 25 questions and answers long.

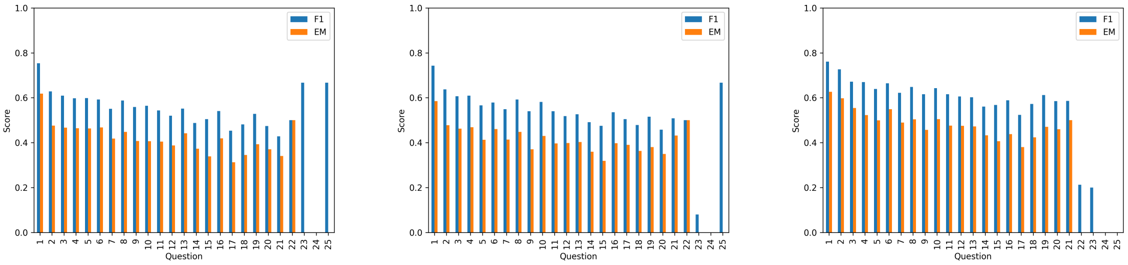


Figure 3: F1 and EM for the 8B model for No DP, A/C DP, and SRL DP

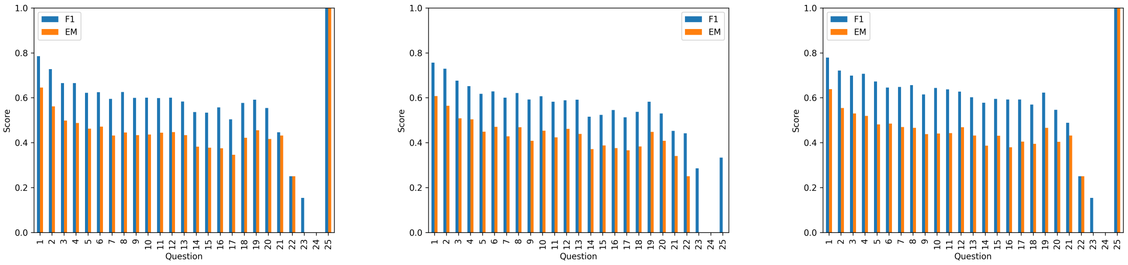


Figure 4: F1 and EM for the 70B model for No DP, A/C DP, and SRL DP

6 Conclusion

From our experiments, we conclude that Dense Paraphrasing, specifically Anaphora and Coreference Dense Paraphrasing (A/C DP) and Semantic Role Labelling Dense Paraphrasing (SRL DP), can help LLMs perform extractive tasks such as QA. A/C DP and SRL DP both enrich texts with semantic information that language models can use to more accurately perform downstream extractive tasks.

Limitations

Our experiments and evaluation were only limited to the CoQA dataset. These texts are all of a similar, finite length. The dataset only covers a limited number of domains and is only in English.

Ethics Statement

Any risks related to the unsupervised use of LLMs are present here. We do not perform manual or automatic checks or filters on the data we have evaluated or in our system. While there are safeguards in place in LLMs to protect from offensive content and bias (Liang et al., 2021; Roy et al., 2023; Sahoo et al., 2024), they are not perfect (Wang et al., 2024). During our limited human evaluation, we did not come across any biased, harmful, or offensive content in the dataset or generated by our system.

Acknowledgements

This work was supported in part by NSF grant 2326985 to James Pustejovsky. The opinions and views reported herein are those of the authors alone.

References

AI@Meta. 2024. Llama 3 model card.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. VerbNet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163, Florence, Italy. Association for Computational Linguistics.

Susan Windisch Brown, Julia Bonn, Ghazaleh Kazeminejad, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2022. Semantic representations for nlp using verbnet and the generative lexicon. *Frontiers in artificial intelligence*, 5:821697.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Peter Bruza and Simon J. Dennis. 1997. Query reformulation on the internet: Empirical data and the hyperindex search engine. In *RIAO Conference*.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *International conference on machine learning*, pages 4558–4586. PMLR.

James Gung. 2020. *Abstraction, Sense Distinctions and Syntax in Neural Semantic Role Labeling*. University of Colorado at Boulder.

James Gung and Martha Palmer. 2021. Predicate representations and polysemy in VerbNet semantic parsing. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 51–62, Groningen, The Netherlands (online). Association for Computational Linguistics.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*.

Gangwoo Kim, Sungdong Kim, Kang Min Yoo, and Jaewoo Kang. 2022. Towards more realistic generation of information-seeking conversations.

Siheng Li, Cheng Yang, Yichun Yin, Xinyu Zhu, Zesen Cheng, Lifeng Shang, Xin Jiang, Qun Liu, and Yujiu Yang. 2023. AutoConv: Automatically generating information-seeking conversations with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1751–1762, Toronto, Canada. Association for Computational Linguistics.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Lluís Màrquez, Xavier Carreras, Kenneth C Litkowski, and Suzanne Stevenson. 2008. Semantic role labeling: an introduction to the special issue.

Anand Rajaraman, Yehoshua Sagiv, and Jeffrey D. Ullman. 1995. Answering queries using templates with binding patterns (extended abstract). In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '95, page 105–112, New York, NY, USA. Association for Computing Machinery.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The coreference under transformation labeling dataset: Entity tracking in procedural texts using event models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

Sarthak Roy, Ashish Harshvardhan, Animesh Mukherjee, and Punyajoy Saha. 2023. Probing LLMs for hate speech detection: strengths and vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6116–6128, Singapore. Association for Computational Linguistics.

Nihar Ranjan Sahoo, Ashita Saxena, Kishan Maharaj, Arif A. Ahmad, Abhijit Mishra, and Pushpak Bhattacharyya. 2024. Addressing bias and hallucination in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 73–79, Torino, Italia. ELRA and ICCL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ziyi Shou, Yuxin Jiang, and Fangzhen Lin. 2022. AMR-DA: Data augmentation by Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3082–3098, Dublin, Ireland. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye, and James Pustejovsky. 2023. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 39–49, Nancy, France. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, and James Pustejovsky. 2022. Competence-based question generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1521–1533, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024. Do-not-answer: Evaluating safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 896–911, St. Julian's, Malta. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Jindou Wu, Yunlun Yang, Chao Deng, Hongyi Tang, Bingning Wang, Haoze Sun, Ting Yao, and Qi Zhang. 2019. Sogou Machine Reading Comprehension Toolkit. *arXiv e-prints*, page arXiv:1903.11848.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*.

A Appendix

	No DP		A/C + SRL DP		SRL + A/C DP	
	EM	F1	EM	F1	EM	F1
Llama3 8B	43.3	57.0	40.8	55.6	40.1	52.4
Llama3 70B	45.9	61.9	45.1	60.9	39.1	54.6

Table 4: CoQA results of text enriched with multiple forms of DP

We also ran the QA task using text that has been enriched by both SRL DP and A/C DP. Including both types of information in the same format hurt performance on the CoQA dataset regardless of the order they were added in. These results are shown in Table 4

Introducing `wwm-german-18k` – Can LLMs Crack the Million? (Or Win at Least 500 Euros?)

Matthias Aßenmacher[1,2†], **Luis Karrlein**[1†], **Philipp Schiele**[3], **Christian Heumann**[1]

[1]Department of Statistics, LMU Munich, [2]Munich Center for Machine Learning (MCML),
[3]Stanford University, Department of Electrical Engineering
Correspondence: matthias@stat.uni-muenchen.de [†] Equal contribution

Abstract

Language-specific evaluation of large language models (LLMs) for multiple-choice question answering (MCQA) is an important means to test their abilities for a multitude of different dimensions. With a data set assembled from questions from the German variant of "Who Wants to Be a Millionaire?" we evaluate a set of German models and ChatGPT concerning factual/commonsense knowledge, syntactic abilities, and logical reasoning, amongst others. We contribute this new MCQA data set, extracted from the show's episodes and designed to evaluate the ability of models to answer this diverse range of questions. To ensure data quality, we describe our preprocessing, encompassing data cleaning, deduplication, and the creation of stratified splits. Furthermore, we fine-tune a set of German LLMs and prompt ChatGPT to provide baseline results. Our findings reveal that these models achieve (partly) satisfactory performance on questions of lower difficulty levels ($\leq$ 1000 euros). As the difficulty increases, performance steadily declines, highlighting the challenging nature of the later stages of the game. We contribute to the ongoing efforts to advance the capabilities of LLMs in comprehending and answering questions by providing a valuable resource for German MCQA research as well as further insights into the limitations of current LLMs.

1 Introduction

Recent advancements in transformer-based language models (Vaswani et al., 2017), especially with the advent of generative large language models (LLMs), such as OpenAI's GPT-series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023), have demonstrated remarkable proficiency in various natural language generation and understanding tasks (Bubeck et al., 2023), including question answering (QA). LLMs are trained on vast amounts of text data from diverse sources, enabling them to learn language patterns, lexical semantics, and seemingly also factual knowledge. The exact extent to which e.g. factual knowledge is present in LLMs (and where exactly it is "stored" in the model weights) is still an open research question to be answered (Meng et al., 2022). As a result of the extensive pre-training, they exhibit impressive capabilities to apparently "comprehend" and respond to a broad spectrum of questions, making them potentially suitable candidates for tackling the challenging task of answering questions from quiz shows like "Wer wird Millionär?" (WWM; English: "Who Wants to Be a Millionaire?").

The QA task in the context of WWM represents an intriguing real-world use case for LLMs due to several compelling factors. First, this task requires not only the comprehension of questions but also the ability to reason, analyze answer choices, and make informed decisions. Second, when investigating the difficulty levels separately, we might gain more insights into how well LLMs can cope with different types of questions, either targeting solely factual knowledge or requiring more complex reasoning abilities. Moreover, the WWM format features questions across a wide range of domains, spanning from commonsense knowledge to more specific fields like science, sports, and pop culture. Consequently, an LLM capable of effectively answering such diverse questions must exhibit world knowledge, as well as factual accuracy, and must be able to grasp linguistic nuances across various topics. Thus, evaluating LLMs on this specific task can shed light on their knowledge representation capabilities and potential to handle multifaceted information.

Contributions: In this paper, we aim to investigate the feasibility and efficacy of employing German fine-tuned LLMs and ChatGPT for answering questions from the WWM quiz show. Our contribution is two-fold:

- We introduce a new multiple-choice question-answering (MCQA) resource for the German language allowing for a more comprehensive evaluation of German LLMs on this task. We (i) gather the data, (ii) extensively describe and motivate the pre-processing steps we applied, and (iii) provide a comprehensive descriptive analysis of the data.

- We evaluate the capabilities of different publicly available LLMs for this task and compare their performance across difficulty levels. This provides a reasonable baseline to compare against when evaluating ChatGPT on this task, even more so when generative LLMs with satisfactory capabilities for German emerge. Comparing fine-tuned to generative LLMs concerning their strengths and limitations in this context, we aim to contribute to the broader understanding of their capabilities and potential real-world applications in QA and game show formats.

2 Related Work: Other MCQA data sets for the German language

To the best of our knowledge, similar data sets from quiz shows or even "Who Wants to Be a Millionaire?" shows in other languages do not yet exist. When on the other hand considering language-specific related work and thus filtering the huggingface datasets space simultaneously for *German* and the task including "*multiple-choice-qa*", there are only eleven resulting data sets as of April 23, 2024.[1] All of these search results are, however, multilingual data sets and thus only a portion of the observations are in German. Other data sets in the German language in the realm of QA exist rather for the task of extractive QA[2], with `deepset/germanquad` (Möller et al., 2021) and `deepset/germandpr` (Möller et al., 2021) being probably the most prominent (purely German) examples. Nevertheless, none of these data sets is specifically aimed at evaluating German models *and* simultaneously targets MCQA. This stresses the need for a new data set for evaluating the ever-improving capabilities of modern-day LLMs.

3 The "wwm-german-18k" Data

3.1 Data Collection

The gathered data originates from the online version of the German quiz show "Wer wird Millionär?" (English: "Who Wants to Be a Millionaire?"), a format that is known across multiple languages. This iconic TV show, celebrated as one of Germany's premier and most recognized programs, challenges contestants with a series of fifteen questions. As they navigate through these questions, they stand a chance to win escalating monetary rewards, peaking at the coveted million Euro prize. These questions span a broad spectrum, from scientific inquiries to pop culture trivia, each of them accompanied by four potential answers and a constrained response time. As the quiz progresses, the complexity of the questions intensifies, but contestants are aided by specific lifelines, known as "Jokers", to facilitate their decision-making. To gather the data, we utilized web scraping techniques to engage with the online version of "Wer wird Millionär? Trainingslager"[3] (English: "WWTBAM? Training Camp"), hosted on RTL's website, the channel that airs the show in Germany. We initiated a game session by sending a POST request to the game's API. After establishing the session, we simulated individual games. For each game, we began at the first level corresponding to the 50 Euro prize. A random question for this level was then presented. Our script recorded the question along with its four possible answers. Importantly, regardless of the answer submitted, the system returned the correct one. This behavior aligns with the game's mechanics, where players are shown the right answer whether or not their guess was accurate. We then added this correct answer to our recorded data. Crucially, the game's training camp structure permits advancing to subsequent levels irrespective of the accuracy of the previous answer, ensuring a new question for each of the game's 15 levels can be drawn in each iteration. As we simulated numerous games, only new questions and their answers were added to our database. Given the assumption that questions are drawn independently, acquiring questions for each level mirrors the coupon collector's problem, where the goal is to collect all unique n items through m draws. We persisted in this iterative approach until reaching a point where new questions rarely emerged, sug-

---

[1] Search results as of April 23, 2024

[2] Search results as of April 23, 2024

[3] `https://spiele.rtl.de/spiele/rtl-spiele/wwm`

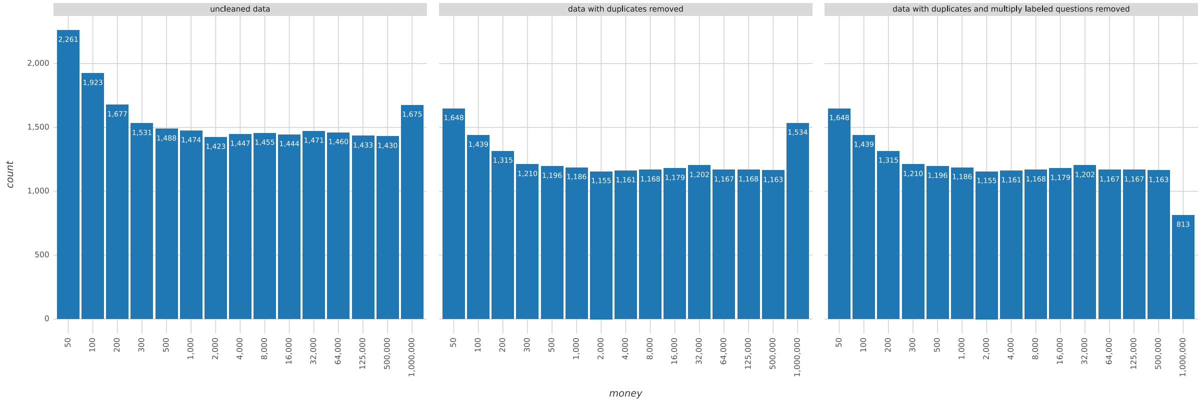


Figure 1: Comparison of the difficulty distribution in the different stages of processed data.

gesting we had captured the majority of available questions. We thus do not claim to have obtained an exhaustive collection of all questions, but rather a substantially representative collection of them that can be considered suitable for evaluating LLMs' capabilities.

3.2 Data Preparation

The unprocessed web-scraped dataset consists of 23,592 questions alongside all four possible answers with the right one as the "label" of the observation. Despite trying to avoid this during web scraping, there was a substantial amount of duplicates in the initial data, i.e. combinations of questions and answer options that occurred multiple times at different prize levels. For these data points, we applied deduplication and assigned the mean of the prize money categories to the new point, rounding down to a tiebreak; so if a data point occurred initially in the second and fifth category it would be put into the third category, and its duplicates removed.[4] Another difficulty encountered in the raw web-scraped data is that some questions came with multiple labels, i.e. multiple correct answer options. As this does not occur in the quiz which the data was taken from, where only one answer at a time is correct, these data points were deleted. These questions with multiple labels were disproportionally frequently present in the Million Euro questions which can be observed when comparing the three distributions in Figure 1. After discarding the erroneous data points, 18,169 deduplicated questions which we deem to be labeled correctly remain. As the final steps of the preprocessing, we sanitized the question endings as they included irrelevant (escape) characters, such as "\n" or spaces at the end of a question. We further unify occurrences of non-standard ellipses ("..", ". . . ", "....") to a common form ("...") for the questions that have to be completed by the quiz show candidate. In three cases, we added missing question marks to the end of a question. The data also contained observations without spaces or ellipses at the end, however, these weren't grammatically complete sentences, but also required completion by one of the answer options. We thus keep them as they were.

Figure 2 illustrates the distribution of the endings, since the context is not always an actual question, but can also be an unfinished sentence that has to be completed. Ellipses ("...") or no ending ("") means the sentence is "cut off" and has to be completed by one of the options. Still, we observe that the majority of the "questions" are actual questions concluded by a question mark. The endings were extracted by using the following regular expression:

```
"(?<=[\w ÄaÖöÜüß%€$+-])[^\w ÄaÖöÜüß%€$+-]*$"
```

From descriptively analyzing the lengths[5] of the questions (i.e. the most important part of the model inputs) we learn that the distribution is notably skewed to the right (cf. Fig. 3). This is reflected by an arithmetic mean of the question length ($\bar{x} = 10.33$), exceeding the median question length ($\tilde{x}_{0.5} = 10$) by a margin of 0.33. While

---

[4] We acknowledge that this is somewhat heuristic, but after careful consideration, we think that this is an acceptable trade-off between the biases of either considering questions as too easy or too hard.

[5] We measure the length in *words*, since there is no uniform definition of tokens and lengths would thus differ from model to model.

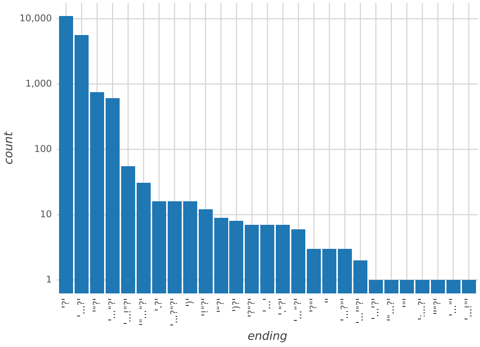


Figure 2: Absolute occurrence frequencies of the context endings in the data set.

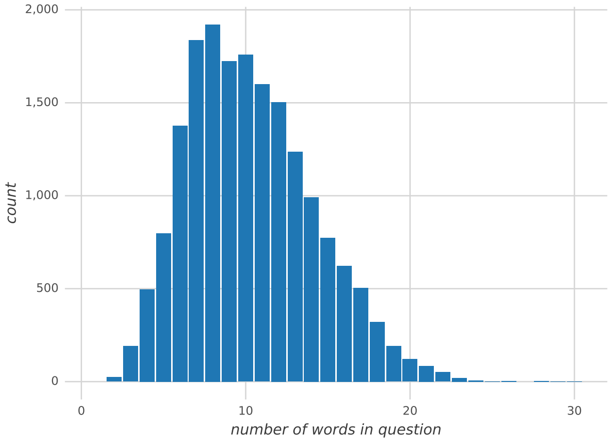


Figure 3: Distribution of the question lengths measured in words. Lengths measured in tokens may differ based on the specific model's tokenizer.

the majority of the questions exhibit a length of less than 10 words, less than 1% of them exceed a length of 20 words.

When subsequently analyzing the lengths of the different possible answers that are provided to the model alongside the question, we take on two perspectives: In Figure 4 we compare the lengths of all wrong answers to the respective correct answers. As there is virtually no difference between the two different distributions we can conclude that models will most likely not be able to pick up on such simple spurious patterns for solving the task. This observation is supported by the result of a Kolmogorov-Smirnov test exhibiting a p-value of $p = 0.95638$, leading to not rejecting the null hypothesis that the distributions do not differ.[6] Figure 4 reveals that the vast majority (approximately 59%) of the answers are only made up of a single word.[7] Nevertheless, there is also a substantial fraction of the answers consisting of two or more words (approximately 41%).

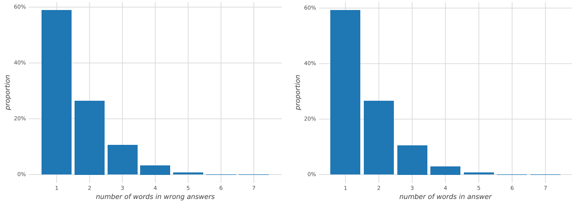


Figure 4: Lengths of wrong (left) and correct answers (right) aggregated across all difficulty levels.

The questions are split into 15 different strata, determined by the amount of money that can be achieved by answering them correctly (cf. Fig. 1 and Tab. 1). With increasing amounts of money to be won, the questions rise in difficulty: Starting from relatively simple ones, or most of the time even just simple puns, for winning 50€, the difficulty increases to ones that are presumably hardest to answer for achieving the 1,000,000€ prize. While for the simple (and often somewhat funny) puns for the first couple of difficulty levels frequently only one possible answer makes sense from e.g. a grammatical point of view, questions at higher levels in the quiz show tend to require either specific knowledge on niche topics or contain multiple plausibly-sounding answers to questions that sound relatively easy at first sight. One example of the latter is shown in Figure 5, where factual knowledge about wine is required. On the other hand, Figure 6 shows a question from the field of physics for a value of only 500 Euros. Here we observe that the answers may also relate to each other. In this case, they all have a connection to the concept of laziness or sluggishness (in German "*Trägheit*") which coincides with the German word for momentum (and is thus the correct answer). Another notable type of question occurs when the answer options themselves allow for excluding options (cf. Fig. 7). In this case three of the four options are distortions of words synonymous with offending or insulting a person, while only the second one, despite also *looking like* a distortion, is a really existing, though rather unknown, word with a different meaning.[8] In the quiz show, contestants try to answer one question from each prize money

---

[6] While this is no statistical proof that the distributions are equal, we at least were not able to show that the distributions differ significantly.

[7] Note, that this might result in more than one *token*, depending on the respective model's tokenizer.

[8] In the case of this particular option, this also certainly re-

money	question	label
50	Worauf radelt man zu zweit?	Tandem
100	Wer viel zu tief ins Glas geschaut hat, ist ...?	hackedicht
200	Teure Restaurants sind oft ...?	piekfein
300	Muss man löhnen, heißt es umgangssprachlich "Zahlemann und ..."?	Söhne
500	Ist es mit der Tugend nicht weit her, spricht der Volksmund von "Sodom und ..."?	Gomorrha
1,000	Was ist fein und glatt und neigt leider häufig zum Nachfetten?	Spaghettihaar
2,000	Wie wird in der Musikszene ein Auftritt bei einem Pop- oder Jazzkonzert genannt?	Gig
4,000	Welche Großstadt liegt nicht in Australien?	Auckland
8,000	Ist in den Medien vom Heppenheimer die Rede, dann geht es meistens um ...?	die Formel 1
16,000	Lindau am Bodensee liegt in ...?	Bayern
32,000	Wer soll gemeinhin mit sogenannten Genussscheinen angelockt werden?	Geldanleger
64,000	Was gewann Andreas Kofler gleich zu Beginn des Jahres 2010?	Vierschanzentournee
125,000	Was sieht aus wie ein Kolibri, ist aber ein Schmetterling?	Taubenschwänzchen
500,000	Bis 1958 hieß das Frisbee ...?	Pluto-Platte
1,000,000	Wo wurde Rainer Maria Rilke 1875 geboren?	Zürich

Table 1: Exemplary questions for each of the 15 difficulty levels.

class at a time, attempting to advance to the very last question to win the million euros. What is, however, of primary interest to us, is not whether an LLM could win the show, but how well it performs per prize money group. We thus perform a stratified train/validation/test split (80%/10%/10%) which ensures a similar balance of all 15 prize money categories across all three splits. For obtaining our baseline performances we utilize the validation set solely for early stopping during fine-tuning, while the test set allows for unbiased testing of the fine-tuned models.

4 Model Evaluation

4.1 Multiple-Choice Question-Answering

MCQA represents a pivotal challenge in natural language understanding and for the probing of factual knowledge. This task requires models to comprehend textual information from the given context or question and to select the most appropriate (i.e. the correct) answer option from a set of given choices, closely mimicking human reasoning and language comprehension abilities. MCQA plays a crucial role in many applications, ranging from exams in education and other assessment systems to information retrieval and chatbots. The complexity of this task lies not only in understanding the nuances of the question and the answer choices but also in grasping the context and potential ambiguities inherent in natural language. In quiz shows, these nuances and ambiguities can be ascribed to a pivotal role since questions or answer options are frequently intentionally created in a way that might confuse the contestant to some extent. Over the years, MCQA has evolved into a multifaceted research problem with applications to various domains (Hendrycks et al., 2020; Pal et al., 2022), incorporating various subtasks such as reading comprehension and semantic, logical (Srivastava et al., 2022), mathematical (Hendrycks et al., 2020), or commonsense (Srivastava et al., 2022) reasoning.

4.2 Model architectures

In recent years, the field of MCQA has witnessed a remarkable transformation due to the advent of (generative) LLMs. There is a clear distinction between autoencoders, relying on discriminative fine-tuned task-specific modeling heads (such as BERT), and generative models that do not necessarily require fine-tuning (such as models from the GPT series). In our work, we rely on six German representatives of the former class of models, providing reasonable baseline values due to their proven and widely examined performance. Huggingface (Wolf et al., 2020) offers access to a wide range of pre-trained architectures via their model hub and allows for seamlessly integrating task-specific heads into the initial model architecture. For this analysis, we chose to use `bert-base-german-cased`, `bert-base-german-dbmdz-cased`, `bert-base-german-dbmdz-uncased`, `deepset/gbert-base`, `german-nlp-group/electra-base-german-uncased`, and `deepset/gelectra-base` alongside the `AutoModelForMultipleChoice` class from Huggingface. While BERT models (Devlin et al., 2019) represent the first large class of fine-tuned task-specific LLMs, ELECTRA (Clark et al., 2020) offers an alternative approach to pre-training, by

---

quires factual knowledge for humans. For well-trained LLMs, however, this could be easier as they might have seen the non-distorted word during pre-training.

context:	Welche dieser Rebsorten ist Grundlage für renommierte Rotweine?	money:	32,000
options:	A: Cabernet Sauvignon, B: Chardonnay, C: Pinot grigio, D: Riesling	answer:	A

Figure 5: Question on the topic of winemaking

context:	Isaac Newton beschäftigte sich intensiv mit dem Prinzip der ...?	money:	500
options:	A: Müdigkeit, B: Bettruhe, C: Trägheit, D: Faulheit	answer:	C

Figure 6: Question from the field of physics

context:	Was macht eine Segelyacht, wenn sie sich zur Seite neigt?	money:	64,000
options:	A: peleidigen, B: krängen, C: spodden, D: ernietrigen	answer:	B

Figure 7: Question with non-words as options

focusing on token-level replacements. Pre-training BERT is mostly focused on the masked language modeling (MLM) task, where a percentage of 15% of the input tokens are corrupted and have to be subsequently predicted by the model. ELECTRA on the other hand employs the MLM task just as an intermediate step performed by an auxiliary generator model which creates predictions for the corrupted tokens and thus returns an ordinary text sequence. The actual ELECTRA model (the discriminator part of the training regime) takes the generator output as an input and is trained to predict for every token whether it is original or produced by the generator. Both models were initially proposed and trained for the English language, but relatively shortly after their release (purely) German versions for both architectures became available. We further examine the performance of ChatGPT (based on GPT-3.5 OpenAI, 2022) as one prominent representative of the class of generative LLMs.

4.3 Experimental Results

Our evaluation mostly focuses on providing reasonable baseline results for future research and differentiating model performance between the different difficulty levels among the questions. When comparing all of the six fine-tuned models and ChatGPT across difficulty levels (cf. Fig. 8 and 9), we observe the expected, relatively steady decline with increasing difficulty of the question (according to the prize money category) for BERT and ELECTRA (cf. Fig. 8), while ChatGPT exhibits constantly better performance for levels other than 1,000,000€ (cf. Fig. 9). Further, despite the overall performance decrease being rather consistent on average, there are still some irregularities. For some fine-tuned models, performance increases for one or two categories at some point on the difficulty scale, but without a clear pattern, and for the 300€ category there is a visible increase in performance *for all BERT/ELECTRA models* compared to the previous category. Overall it is important to keep in mind that an accuracy of 25% corresponds to random guessing, which is on average (nearly) the case for most of the higher prize money categories (also for ChatGPT). Concerning a comparison of the fine-tuned architectures, BERT vs. ELECTRA, models of the latter architecture (i) exhibit a higher average accuracy across all different difficulty levels which (ii) can be explained by better performance on especially the low-difficulty categories (below 2,000€). Model performance of ChatGPT proves to be very stable and high across most difficulty levels before it eventually starts to notably decrease at 64,000€ and exhibits a sharp drop for 1,000,000€. For the easier questions, there are only a few differences between the different fine-tuned models of the two underlying architectures, and we also do not observe notable differences to the performance of ChatGPT.[9] Nonetheless, we observe an overall performance difference between the two fine-tuned ELECTRA models. While deepset's `deepset/gelectra-base` achieves an accuracy of 53.83%, german-nlp-group's `german-nlp-group/electra-base-german-uncased` is better by a margin of 3.74% with a 57.57% accuracy. Another interesting observation is the decreasing variability in the accuracy with increasing question difficulty, thus decreasing overall model performance.

5 Discussion

Arguably, we do not (yet) use the data set to its full potential in this set of experiments, since we

---

[9] Nevertheless, one needs to keep in mind that ChatGPT is neither fine-tuned nor provided with few-shot examples.

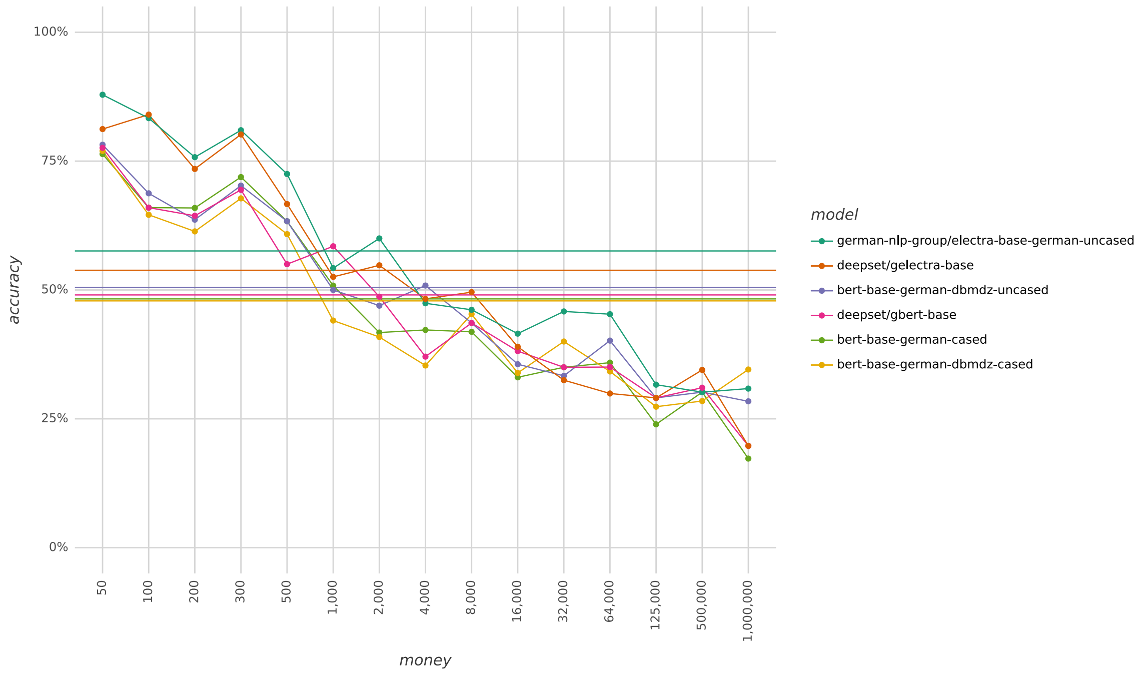


Figure 8: Accuracy of the evaluated models that were fine-tuned from different open-source models (separated by colors) split by the difficulty level of the questions (x-axis). Horizontal lines (in the respective colors) represent the model's average performance values across *all* difficulty levels.

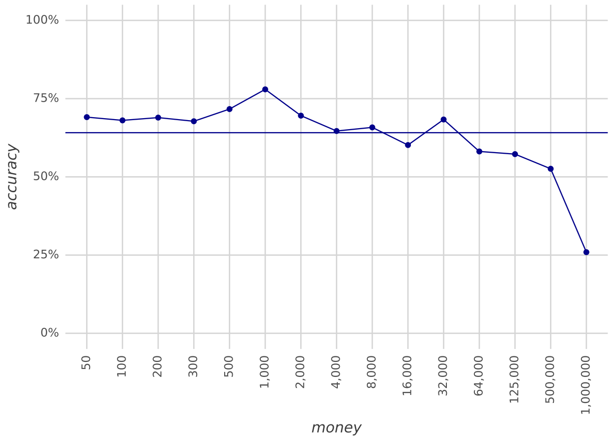


Figure 9: Accuracy of ChatGPT split by the difficulty level of the questions (x-axis).

only investigate the capabilities models can either *acquire* when fine-tuned on this MCQA data or the capabilities a generative LLM *already exhibits*. Beyond this use case, there's further potential for using the data in various few-shot learning settings, allowing for a more in-depth evaluation of prompting-based or generative LLMs. This few-shot setting would test the world knowledge and the reasoning capabilities already present in the LLMs' model weights, whilst nudging the model in the right direction, thus taking on another angle on this problem set. The results obtained from this evaluation of German BERT, ELECTRA, and ChatGPT on our newly introduced `wwm-german-18k` data set, however, still provide valuable insights into the strengths and limitations of heavily used LLMs in handling this large and diverse set of questions with varying levels of difficulty. The remainder of this discussion section will nevertheless shift the focus to potential enhancements that (open-source) generative LLMs can bring to solving MCQA tasks, along with a critical examination of the data set's potential shortcomings.

Recent developments in generative LLMs have led to remarkable performance when it comes to (seemingly) understanding and generating natural language text, which could also turn out to be a notable advantage for MCQA tasks. In response to closed-source models like ChatGPT, new generation of LLMs that is first and foremost characterized by openly-available weights emerged, preliminarily culminating in the publication of Llama3 (Meta, 2024) on April 18, 2024. Besides the "base" versions, many of these models are also released as quantized, instruction-tuned, or mixture-of-experts versions allowing for (a) computationally cheaper adaption and (b) seamless usage of the models. Such models could potentially simultaneously benefit from the training example while exhibiting all the advantages that generative LLMs have over discriminative ones. This flexibility may empower them to also excel in tasks beyond MCQA, where answer choices are not explicitly provided, or when questions require generating more nuanced and contextually relevant responses. Additionally, generative LLMs could be leveraged for data augmentation purposes or for generating new, additional questions, thus enhancing the diversity and complexity of the data set. However, despite the diverse and interesting setting this data set is placed in, several potential shortcomings should also be acknowledged. The questions in the game show, and hence in the data set, may contain pop culture references, idiomatic expressions, or very specialized knowledge, which can pose challenges for both generative and discriminative models, especially when applied to more general domains. Additionally, the data set's focus on factual knowledge and trivia may not be fully adequate to evaluate the models' abilities in understanding and reasoning about more abstract or complex concepts holistically, which are arguably rather important for real-world applications.

Summing up, these debatable discussion points underscore the need for adequate resources to evaluate the promise of generative LLMs advancing the capabilities on MCQA, amongst others. The introduction of the `wwm-german-18k` data depicts an important step in that direction due to its challenging nature, for machine learning models and for humans. Simultaneously we also want to highlight the need for further data sets encompassing a broader range of question types and domains to further evaluate and refine these models. Future research needs to further aim at developing more diverse and contextually rich MCQA data sets that better represent the complexities of natural language understanding, ultimately driving the development of such data sets close to real-world scenarios will help to robustify LLM systems for MCQA across various languages and domains.

6 Conclusion

In conclusion, this research presents a dedicated and well-curated contribution to the field of German MCQA based on data extracted from the popular TV show "Wer wird Millionär?" alongside important baseline results for future research, showcasing one of the intended uses of the data: Evaluation of the progressing capabilities of LLMs. The primary contributions of this study can thus be summarized as follows:

First and foremost, we introduce a novel MCQA data set for the German language derived from the German version of the show "Who Wants to be a Millionaire?". This data set encompasses approximately 18,000 observations and thus provides a valuable resource for evaluating a diverse set of capabilities ascribed to modern-day LLMs. The diverse range of questions in the data spans various dimensions from factual/commonsense knowledge, over syntactic abilities to logical reasoning. To ensure the quality and reliability of our dataset, we carefully describe the careful preprocessing steps we took, which involved several aspects of cleaning the data, question deduplication, and the creation of stratified data splits. These steps are crucial for maintaining high data quality and providing a foundation for further research.

We also conducted extensive experiments using fine-tune two state-of-the-art German language models, namely German BERT and ELECTRA, as well as ChatGPT on our data set. The obtained baseline results offer insights into the performance of LLMs on this task, highlighting their competence in addressing questions with lower difficulty levels, up to approximately 1000€. However, as question complexity increases, our results reveal a consistent decrease in model performance, shedding light on the challenging nature of more difficult questions. This finding underscores the need for further research and model development to address these challenges and enhance MCQA performance on complex questions. Eventually, our contributions in the form of a new German MCQA dataset, detailed preprocessing methodology, and baseline results provide a valuable new resource for advancing the capabilities of German LLMs in comprehending and answering questions in natural language, particularly within the context of popular culture and entertainment. Further, it might serve as a blueprint for other languages, as this game show is popular around the world. This work invites researchers to build upon our findings and explore innovative approaches to improve the robustness and accuracy of MCQA models, ultimately contributing to the development of more robust and capable LLM-based systems.

Limitations

While we hope that this work provides researchers with a valuable non-English language resource for a more diverse evaluation of LLMs to gain more nuanced insights into their strengths and weaknesses, there are still issues we do not yet address in this work: First, we do not provide an exhaustive evaluation and comparison of different (open- vs. closed-source) generative LLMs, since this is not the focus of this work. Our focus is on the introduction of this new resource for comparing and evaluating LLMs. Second, this resource can also only be seen as a small contribution to the bigger question of how to properly benchmark generative LLMs. It only covers certain aspects of language and culture, but we hope this can serve as a valuable contribution to a better understanding of LLMs' behavior. Finally, as described in Section 3.2 there were some inconsistencies when recording the prize money category during web scraping, which we attribute to the subjectiveness of the concept of "difficulty" in the realm of quiz show questions. We would thus argue that our method for assigning the category can be regarded as a realistic approximation of the average perceived difficulty.

Ethical Considerations

We affirm that our research adheres to the ACL Ethics Policy. This work involves the use of data that is publicly available on the internet and does not involve harmful content or any personally identifiable information on humans. We declare that we have no conflicts of interest that could potentially influence the outcomes, interpretations, or conclusions of this research. All funding sources supporting this study are acknowledged in the acknowledgments section. We have made our best effort to document our methodology, experiments, and results accurately and are committed to sharing our code, data, and other relevant resources to foster reproducibility and further advancements in research.

Acknowledgements

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI - grant number 460037581.

Availability of the Data

Unfortunately, we are not able to share the data due to legal considerations: While scraping the data can, to the best of our knowledge, be considered legally fine, making it available to the public might lead to violations of copyright. We made several attempts to contact the production company EndemolShine Germany, explaining our endeavor to cooperate with them for publishing the data for research purposes only, but they were not open to any sort of conversation about this. Our legal council subsequently advised us against publishing the data without the company's explicit consent. Interested researchers are, however, encouraged to contact us for obtaining access to the data set.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval. *arXiv preprint arXiv:2104.12741*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Native Language Identification Improves Authorship Attribution

Ahmet Yavuz Uluslu and **Gerold Schneider**
University of Zurich
{ahmet.uluslu, gerold.schneider}@linguistik.uzh.ch

Can Yildizli
PRODAFT
can@prodaft.com

Abstract

This study investigates the integration of native language identification into authorship attribution, a previously unexplored aspect that is particularly important in multilingual contexts. We introduce AA-NLI50, a new dataset containing both native language and authorship information. We propose a novel chain-of-thought approach for native language identification. Our findings demonstrate that our system significantly enhances authorship attribution performance, with results showing a mean accuracy improvement of 9% over baseline methods.

1 Introduction

Authorship attribution—the task of identifying the author of a given document based on a set of possible candidates—is crucial in various forensic applications (Koppel et al., 2009). The complexity of the task increases significantly with the number of potential candidates and the scarcity of training data (Luyckx and Daelemans, 2011; Rocha et al., 2016). Most recent studies integrated author profiles that include attributes such as gender, education level and age to refine the attribution process and narrow down the pool of suspected authors (Deutsch and Paraboni, 2023). Despite these advances, the impact of native language in authorship analysis remains largely unexplored, even though it is frequently mentioned in forensic applications, particularly in the context of cybercrime (Perkins, 2021).

Native Language Identification (NLI) is the task of automatically identifying the native language (L1) of an individual based on their writing or speech in another language (L2). The underlying hypothesis is that L1 affects L2 production due to cross-linguistic influence (Yu and Odlin, 2016). Recent findings in neuroscience suggest that structural differences in the brain can result from the influence of the native language (Wei et al., 2023).

The majority of NLI research relies on learner corpora, such as TOEFL11 (Blanchard et al., 2013) and ICLE (Granger et al., 2009). The training examples consist of formal writings in classroom settings that differ considerably from the context and register of ransomware notes or forum posts on the dark web (Jin et al., 2022). The mismatch can have a negative impact on the analysis and affect the overall performance (Grieve, 2023). Previous studies have demonstrated that state-of-the-art NLI systems often struggle to generalize across different topics and linguistic registers (Malmasi and Dras, 2018). While datasets derived from social media platforms such as Reddit (Goldin et al., 2018; Murauer and Specht, 2021) offer a diverse range of samples within an informal register, which helps to mitigate this issue, they still encounter significant challenges related to topic imbalance.

From an architectural standpoint, NLI followed the trend towards attention-based transformation models such as BERT (Steinbakken and Gambäck, 2020), BigBird (Kramp et al., 2023) and GPT-2 (Lotfi et al., 2020). To address practical problems, more recent work has focused on improving scalability (Uluslu and Schneider, 2022) and extending these models to languages other than English (Malmasi and Dras, 2017; Uluslu and Schneider, 2023). The emergence of more powerful large language models such as GPT-4 (Achiam et al., 2023) and Mixtral 8x7B (Jiang et al., 2024) enabled increasing capabilities in zero-shot learning, achieving state-of-the-art performance on various tasks and benchmarks (Chang et al., 2023). Early results in NLI demonstrate the potential to overcome existing limitations, including scalability to more languages, generation of explanations, identification of direct translations, and cross-domain adaptability (Zhang and Salle, 2023). While LLMs provide new capabilities in this field, they also introduce significant challenges related to robustness, as well as increased risks of hallucinations and biases.

The main contributions and findings of this study are threefold: (1) We create a new dataset called AA-NLI50 which includes both the author's native language and authorship information; (2) We propose a zero-shot chain-of-thought (CoT) approach to mitigate hallucinations found in previous NLI studies; (3) We show that predicting the author's native language significantly improves attribution performance in multilingual contexts.

2 Related Work

In the NLI shared tasks of 2013 and 2017, the best performing approaches primarily used linguistic features together with traditional machine learning algorithms (Tetreault et al., 2013; Malmasi et al., 2017). Various feature types were investigated, including spelling errors, word and lemma n-grams, character n-grams, dependency parsing and morphosyntax (Malmasi and Dras, 2018). The combination of these diverse features was shown to be highly effective in achieving the best results for NLI (Markov et al., 2022). More recently, the deep generative approach was introduced, involving the fine-tuning of a GPT-2 model to identify each L1, achieving state-of-the-art performance (Lotfi et al., 2020). However, this approach was found to be resource-intensive with considerable performance bottlenecks (Uluslu and Schneider, 2022). The replication attempts highlighted generalization issues across various domains, suggesting potential overfitting to the benchmark dataset (Vian, 2023). Most recent studies found that newer LLMs, such as GPT-4 (Achiam et al., 2023), achieve state-of-the-art performance in zero-shot settings using prompt-based approaches, which represents a significant advancement over previous methods (Zhang and Salle, 2023; Goswami et al., 2024).

Authorship profiling has been explored as a valuable tool for authorship attribution where it assists in narrowing down the pool of potential candidates by filtering based on characteristics such as gender, age, and educational background (Yang and Chow, 2014; Deutsch and Paraboni, 2023). Psychological profiling was also shown to be effective in differentiating between authors in various contexts (Boyd, 2018). The impact of the author's native language has not yet been explored due to the scarcity of data and specific use cases. The significance of native language in cybercrime investigations cannot be overstated, as evidenced by its repeated utility in forensic analyses (Perkins, 2021).

3 Data

Due to the absence of available authorship attribution datasets that include the native language of the author, we scraped a new dataset from the social media platform Reddit, following the methodologies of Murauer and Specht (2019, 2021); Goldin et al. (2018). The dataset features English posts from authors in five different L1: French (FR), Dutch (NL), Turkish (TR), Russian (RU), and German (DE). We included posts that were assigned the topic *politics*, most of which discuss recent migration and economic issues in Europe. We ensured a minimum of 10 authors for each L1, with each author contributing at least 20 documents. A document is defined as a concatenation of individual posts until the minimum document length is reached. Following the pre-processing steps of Murauer and Specht (2019), we required each document to have a minimum length of 4,000 characters and replaced URLs and user tags with special tokens. The final corpus consists of 1,000 documents in total.

4 Methodology

We build upon the work presented by Deutsch and Paraboni (2023) by incorporating profiling systems to enhance closed-set authorship attribution. This approach utilizes an ensemble architecture comprising word and character-level n-gram models (Custódio and Paraboni, 2019). The output probabilities from the word and character-level n-gram models, combined with the one-hot encoded native language prediction from the LLM, are fed into a second-level logistic regression classifier to identify the author of the input documents. The entire pipeline is illustrated in Figure 1. To select the most suitable language model, we conducted preliminary experiments using the TOEFL11 (Blanchard et al., 2013) dataset, the de facto benchmark for NLI. The results presented in Appendix A.1 show that Llama3 (AI@Meta, 2024) performs better than Mixtral 7Bx8 (Jiang et al., 2024) but is slightly outperformed by GPT-4 (Achiam et al., 2023). Due to the confidential nature of forensic work, we only consider open-source models and utilize *llama3-70b-8192* for our experiments. The discrepancies between GPT-4 and Llama3 were primarily observed in the problematic pair within the benchmark (Hindi-Telugu). Early results in NLI revealed various types of hallucinations (Zhang and Salle, 2023), likely due to the cultural and contextual cues

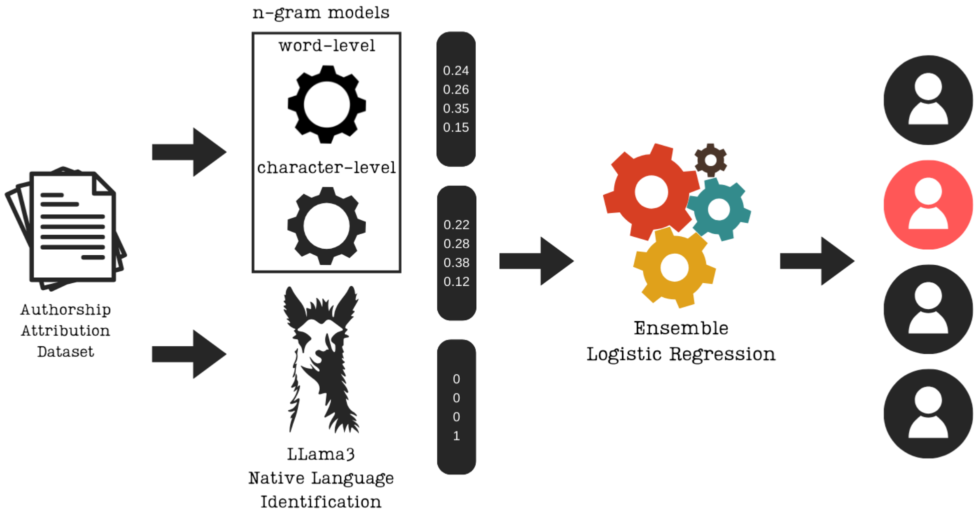


Figure 1: System architecture demonstrating the integration of native language profiling with word and character-level n-grams for authorship attribution.

present in the benchmark. Learner corpora often contain biographical information about the author, which the LLM leverages when trying to identify the native language. Using self-reported information raises the question of whether this constitutes cheating the task (Balloccu et al., 2024), as it can result in cultural analysis rather than understanding the linguistic patterns. This concern aligns with previous findings that model-generated explanations are unreliable indicators of the model's reasoning process (Madsen et al., 2024). The model can hallucinate in order to maintain self-consistency, even in the absence of linguistic cues. To address this issue, we propose a CoT approach, which does not eliminate hallucinations per se, but was shown to restrict the model behavior to the instructions (Dhuliawala et al., 2023). We redacted various name entity information from the text, including geopolitical entities (GPE), locations (LOC), and nationalities or religious or political groups (NORP), which can reveal the author's origins. To enforce a structured analysis and delay the prediction until all instructions are followed, we introduce XML formatting and prefill the model's response. The full system prompt used in our study can be found in Appendix B.1.

5 Results

We first present the results of NLI as an independent task. Following this, we integrate the most realistic setup into the authorship attribution pipeline to derive the final results. This two-step approach ensures that our evaluation captures both the isolated and integrated performance of the models.

5.1 Native Language Identification

We conducted four experiments to evaluate the zero-shot performance of LLMs on NLI. Using the entire corpus, we reported the results in terms of precision, recall, and F1 score. For comparison, we implement the open-set classification prompt from Zhang and Salle (2023) as well as our CoT approach on the dataset. We report results after redacting entity information in both experiments to assess the model dependency. Previous research has reported near-perfect accuracy on benchmarks for most language pairs under zero-shot settings. We argue that these results should be interpreted cautiously, as state-of-the-art approaches typically require approximately 10,000 examples to achieve similar performance levels and often encounter generalization issues across different datasets. While LLMs may exhibit an emergent ability for NLI, it is also possible that they have merely adapted to the datasets, finding shortcuts due to the task's inherent complexity (Schaeffer et al., 2024). An example of such hallucinations can be observed in B.2.

Approach	**P**	**R**	**F1**
Baseline	0.68	0.68	0.69
- Redacted ↓	0.49	0.46	0.48
CoT Prompt (Ours)	0.54	0.53	0.54
- Redacted ↓	0.46	0.46	0.47

Table 1: Performance comparison of open-set classification and CoT approaches before and after redaction.

As shown in Table 1, the open-set classification prompt achieved a high performance of 69% under zero-shot settings on a more complex dataset compared to the benchmark. However, redacting entity information resulted in a substantial performance decrease of 20%. In comparison, the CoT approach requires the model to document its findings before making a final prediction, relying more on the model's ability to identify linguistic features. While the CoT prompt also experienced a performance drop due to the removal of entity information, the decrease was not as significant as with the original prompt. Both approaches converged to a similar level of performance in the follow-up experiments.

# Authors	**Accuracy**	
	Baseline	**+ Native Language**
5	0.65	0.66
10	0.46	**0.55**
15	0.37	**0.49**
20	0.33	**0.43**
25	0.27	**0.42**
30	0.25	**0.34**
35	0.22	**0.33**
40	0.20	**0.29**
45	0.18	**0.28**
50	0.17	**0.27**
Mean	0.32	**0.41**

Table 2: Authorship attribution mean accuracy and SD results for the standalone and integrated pipeline.

5.2 Authorship Attribution

Following the evaluation methodology of Deutsch and Paraboni (2023), we completed multiple evaluation experiments to assess the system's performance under varying conditions. We employed a zero-shot classification system, eliminating the need to split the dataset between attribution and profiling tasks. We created a balanced testing set comprising 20% of the entire dataset (200 documents), including 50 authors and five different L1s. We conducted the experiments using the CoT approach, as it offers a more realistic performance given the absence of self-reported information in forensic contexts. To evaluate the system, we sampled random sets of candidate authors from the 50-author test set, varying the number of candidate authors from 5 to 50. To minimize the effects of random selection, each evaluation was repeated 20 times. For each iteration, we varied the candidate authors randomly and selected different training and testing documents. This repetition aimed to provide more reliable and robust results by averaging out the variability introduced by random selection. The results of the authorship attribution experiments are reported in terms of accuracy, as shown in Table 2. The table presents the mean accuracy scores obtained by the open-set classification baseline and the integration of NLI into the stack ensemble. The best results for each candidate set are highlighted in bold. Based on McNemar's test, the differences in performance between the baseline model and the proposed model were found to be statistically significant ($p < 0.05$) after 5 authors. The results indicate that incorporating native language outperforms the baseline as the number of candidate authors increase. Overall, we achieved a 9% increase in mean accuracy, with the baseline at 33% and the enhanced ensemble model at 41%.

6 Conclusion

Our study demonstrates that integrating native language into authorship attribution systems significantly enhances attribution accuracy, which is particularly important in multilingual contexts such as cybercrime investigations (Perkins, 2021). This improvement aligns with the gains observed from other profiling attributes like age, gender, and education (Deutsch and Paraboni, 2023). Our study highlights the shortcuts taken by LLMs in profiling tasks, with a particular focus on how certain background information in the text (e.g. ethnicity) can lead to superficial analysis and hallucinations. Therefore, we argue that model generations should not be considered true explanations of the reasoning process. We found that employing CoT prompts can mitigate this tendency by encouraging systematic documentation of relevant linguistic features. While these findings offer promising advancements, they also underscore the need for cautious interpretation of LLM outputs in forensic sciences. Future research should focus on developing more robust profiling techniques that account for diverse linguistic factors, including the effects of register, genre, and topic. As LLMs continue to play increasingly important roles in authorship analysis, our work emphasizes the ongoing need to investigate their behaviors and limitations.

Acknowledgments

This work was supported by the collaboration between the University of Zurich and PRODAFT as part of the Innosuisse innovation project AUCH 103.188 IP-ICT (Author profiling to automatize attribution in cybercrime investigations).

References

OpenAI Josh Achiam, Steven Adler, and Sandhini Agarwal. 2023. GPT-4 Technical Report.

AI@Meta. 2024. Llama 3 model card.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013(2):i–15.

Ryan L Boyd. 2018. Mental profile mapping: A psychological single-candidate authorship attribution method. *PloS one*, 13(7):e0200588.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

José Eleandro Custódio and Ivandré Paraboni. 2019. An ensemble approach to cross-domain authorship attribution. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings 10*, pages 201–212. Springer.

Caio Deutsch and Ivandré Paraboni. 2023. Authorship attribution using author profiling classifiers. *Natural Language Engineering*, 29(1):110–137.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Gili Goldin, Ella Rabinovich, and Shuly Wintner. 2018. Native language identification with user generated content. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3591–3601.

Dhiman Goswami, Sharanya Thilagan, Kai North, Shervin Malmasi, and Marcos Zampieri. 2024. Native language identification in texts: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3149–3160.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, Magali Paquot, et al. 2009. *International corpus of learner English*, volume 2. Presses universitaires de Louvain Louvain-la-Neuve.

Jack Grieve. 2023. Register variation explains stylometric authorship analysis. *Corpus Linguistics and Linguistic Theory*, 19(1):47–77.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Youngjin Jin, Eugene Jang, Yongjae Lee, Seungwon Shin, and Jin-Woo Chung. 2022. Shedding new light on the language of the dark web. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5621–5637, Seattle, United States. Association for Computational Linguistics.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26.

Sergey Kramp, Giovanni Cassani, and Chris Emmery. 2023. Native language identification with big bird embeddings. *arXiv preprint arXiv:2309.06923*.

Ehsan Lotfi, Ilia Markov, and Walter Daelemans. 2020. A deep generative approach to native language identification. In *Proceedings of the 28th international conference on computational linguistics*, pages 1778–1783.

Kim Luyckx and Walter Daelemans. 2011. The effect of author set size and data size in authorship attribution. *Literary and linguistic Computing*, 26(1):35–55.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful?

Shervin Malmasi and Mark Dras. 2017. Multilingual native language identification. *Natural Language Engineering*, 23(2):163–215.

Shervin Malmasi and Mark Dras. 2018. Native language identification with classifier stacking and ensembles. *Computational Linguistics*, 44(3):403–446.

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A report on the 2017 native language identification shared task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–75.

Ilia Markov, Vivi Nastase, and Carlo Strapparava. 2022. Exploiting native language interference for native language identification. *Natural Language Engineering*, 28(2):167–197.

Benjamin Murauer and Günther Specht. 2019. Generating cross-domain text classification corpora from social media comments. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings 10*, pages 114–125. Springer.

Benjamin Murauer and Günther Specht. 2021. Dt-grams: Structured dependency grammar stylometry for cross-language authorship attribution. *arXiv preprint arXiv:2106.05677*.

Ria C Perkins. 2021. The application of forensic linguistics in cybercrime investigations. *Policing: A Journal of Policy and Practice*, 15(1):68–78.

Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. 2016. Authorship attribution for social media forensics. *IEEE transactions on information forensics and security*, 12(1):5–33.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

Stian Steinbakken and Björn Gambäck. 2020. Native-language identification with attention. In *Proceedings of the 17th international conference on natural language processing (icon)*, pages 261–271.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 48–57.

Ahmet Yavuz Uluslu and Gerold Schneider. 2022. Scaling native language identification with transformer adapters. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 298–302.

Ahmet Yavuz Uluslu and Gerold Schneider. 2023. Turkish native language identification. *Natural Language and Speech Processing (ICNLSP-2023)*, page 303.

Matias Johansen Vian. 2023. A study of transformers for cross-corpus native language identification. Master's thesis, NTNU.

Xuehu Wei, Helyne Adamson, Matthias Schwendemann, Tomás Goucha, Angela D Friederici, and Alfred Anwander. 2023. Native language differences in the structural connectome of the human brain. *Neuroimage*, 270:119955.

Min Yang and Kam-Pui Chow. 2014. Authorship attribution for forensic investigation with thousands of authors. In *ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings 29*, pages 339–350. Springer.

Liming Yu and Terence Odlin. 2016. *New perspectives on transfer in second language learning*, volume 92. Multilingual Matters.

Wei Zhang and Alexandre Salle. 2023. Native language identification with large language models. *arXiv preprint arXiv:2312.07819*.

A Appendix – Preliminary Experiments

Model	**TOEFL11 Test Set (%)**
Random Baseline	9.1
GPT-2 (Lotfi et al., 2020)	89.0
GPT-3.5 (Zero-shot)	74.0
Mixtral 8x7B (Zero-shot)	74.0
LLama3 70B (Zero-shot)	85.4
GPT-4 (Zero-shot)	91.5

Table A.1: Performance comparison of various models on the TOEFL11 test set.

B Appendix – Supplementary Material

B.1 System Prompt

You are a forensic linguistics expert responsible for analyzing texts written by non-native speakers. Identify linguistic cues such as direct translations, spelling errors, syntactic patterns, and grammatical errors to identify the native language of the author. It is important to note that the self-reported information or cultural references provided in the text can be misleading.

<transcript> {input_text} </transcript>

Think step by step on how to analyze the <transcript> within the provided <sketchpad>.

In the <sketchpad>, return a list of <findings> and their corresponding <types>.

Then, check that <sketchpad> items are factually consistent with the <transcript>.

Finally, identify the native language of the author based on the <sketchpad>.

Figure B.1: System Prompt

B.2 LLM-generated Authorship Profiling Outputs

Text	**Prediction**
If the state wants to implement something bad, they protest like mad men, until the state listens to them. The Gezi protests failed, because we didn't go hard enough. <text continues>	Turkish <hallucination analysis>
If the state wants to implement something bad, they protest like mad men, until the state listens to them. The Moscow protests failed, because we didn't go hard enough. <text continues>	Russian <hallucination analysis>
If the state wants to implement something bad, they protest like mad men, until the state listens to them. The <redacted> protests failed, because we didn't go hard enough. <text continues>	Random or No prediction <complication message> <random message>

Figure B.2: LLM-generated outputs for the NLI task based on Reddit posts under different conditions.

Design and Comparison of Arabic Negotiation Bots Using LLMs versus Seq2Seq Models with Reinforcement Learning

Ahmad Hajj, Yasmine Abu Adla, Samah Albast, Hazem Hajj, Shady Elbassuoni, Wassim El Hajj, Khaled Shaban
University of Wisconsin–Madison, US; American University of Beirut, Lebanon;
Qatar University, Qatar
ahajj@wisc.edu, {yaa41, sma122, hh63, se58, we07}@aub.edu.lb, khaled.shaban@qu.edu.qa

Abstract

Negotiation is a crucial aspect of daily life, spanning from personal agreements to organizational contracts. As AI continues to evolve, the delegation of negotiation tasks to machines is becoming increasingly feasible. However, while research on English negotiation bots is progressing, similar advancements in other languages, particularly Arabic, are lacking. This paper presents the first design of Arabic negotiation bots, employing two approaches. The first approach involves utilizing a pre-trained Large Language Model (LLM) for this task via prompt engineering, while the second leverages Meta's "Deal or No Deal" framework, integrating a sequence-to-sequence (Seq2Seq) model with reinforcement learning (RNN-RL). We introduce and release two new datasets tailored for the models, including LLM prompt instructions and an Arabic dataset for the RNN-RL model. Our experiments demonstrate both the potential and limitations of these approaches, providing essential insights into their performance and outlining future research directions for multilingual negotiation bots.

1 Introduction

Negotiations occur in various contexts, from individuals haggling over a purchase to companies discussing contracts and governments seeking economic advantages (Halver, 2022; Founders, 2022). With the recent advancements in chatbots, negotiation has emerged as an ideal application for artificial intelligence (AI), enabling intelligent agents to negotiate and cooperate effectively.

Research on English negotiation bots has evolved, initially employing game theory and rule-based approaches (Jennings et al., 2001; Hussain, 2014; Balachandran and Mohammadian, 2015; Koley and Rao, 2018), followed by deep learning methods that achieved significant success in natural language processing. Meta's work (Lewis et al., 2017) laid the foundation for product negotiation using two sequence-to-sequence (Seq2Seq) recurrent neural networks (RNN) models representing the buyer and the seller. Those models were trained in a supervised fashion using human-developed negotiation data, with parameters further improved through reinforcement learning (RL). Building upon Meta's RNN-RL models, several other works (He et al., 2018; Wang et al., 2019; Zhou et al., 2019; Mishra et al., 2022; Raut et al., 2023) introduced further improvements, including strategies for emotion, persuasion, and politeness. Furthermore, transformer-based sequence models like generative pre-training transformer (GPT) models for negotiation bots have been developed (Fu et al., 2023). While those English negotiation bots have seen continued advancements, research on non-English negotiation bots, including Arabic, remains unexplored.

To address this research gap, we introduce two approaches to automate non-English negotiations with application to Arabic and create, to the best of our knowledge, the first Arabic negotiation bots. In our first approach, we propose utilizing a pre-trained large language model (LLM), namely GPT-3.5 Turbo, to automate Arabic negotiations through prompt engineering. Moreover, as a baseline, we develop an Arabic negotiation model using a combination of a Seq2Seq RNN with reinforcement learning (RNN-RL), which is inspired by Meta's popular "Deal or No Deal" model (Lewis et al., 2017). To train the RNN-RL model, we create the first Arabic negotiation dataset by automatically translating Meta's English negotiation dataset (Lewis et al., 2017). Our experimental results demonstrate the superiority of the LLM approach based on human evaluation across various metrics such as negotiation coherence, complexity, language quality, style and. dynamics.

In summary, our contributions in this paper include the introduction of the first Arabic negotiation bots along with the following methods and datasets:

- A zero-shot prompt engineering approach that can be used to tailor pre-trained LLMs to automate negotiations.
- A baseline model for Arabic negotiation bots using Seq2Seq with reinforcement learning.
- A dataset of prompts that can be used to instruct pre-trained LLMs to perform negotiations.
- An Arabic negotiation dataset.

2 Related Work

Many approaches have been explored in the development of negotiating bots. Early work employed game theory, focusing on rational actions for self-interested agents. These approaches utilized strategies like monotonic concession, where one or both negotiation parties must concede each time an agreement is not reached, following specific methods to determine the conceding agent and the extent of the concession (Jennings et al., 2001). Incorporating game theory into negotiation considers that the agent must select the best strategy from the space of all possible strategies, which often requires computationally expensive calculations.

Other approaches (Hussain, 2014; Balachandran and Mohammadian, 2015) used rule-based methods, which relied on a set of if-then statements to determine the negotiating agent strategy. These rules dictated how the agent should respond in different situations during the negotiation process. Another traditional approach employed in the development of negotiating agents was the use of heuristics (Koley and Rao, 2018) to guide the decision-making process. One heuristic is based on the observation that there is an inverse relationship between an issue's preference and how frequently its value changes during negotiations. Another heuristic was based on the correlation between the preference for a value and its frequency of being offered. These heuristics help in understanding the opponent's preferences and adjusting the negotiation strategy accordingly. All of the aforementioned approaches to building a negotiating bot rely on a handcrafted set of rules and they lack the capability to truly negotiate using natural language.

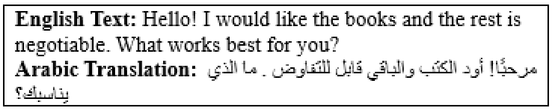
Many recent works used deep learning to develop negotiating bots. The work in (Lewis et al., 2017) is the first to apply deep learning techniques to build an end-to-end model for natural language negotiation. The model learns both linguistic and reasoning skills using supervised and RL techniques. In (He et al., 2018), they addressed the problem of degeneracy of the work in (Lewis et al., 2017) by decoupling the negotiating strategy from language generation. In (Wang et al., 2019), they focused on persuasion strategies using a human-human persuasion dialogue dataset with persuasion strategy annotation. In (Zhou et al., 2019), a negotiation coach provided tactics to help the seller achieve improved deals by employing an LSTM-based model to generate tactic suggestions. In (Mishra et al., 2022), they trained and fine-tuned a LM using RL while considering various sub-rewards for persuasion, emotion, politeness, coherence, and repetitiveness. In (Raut et al., 2023), they developed a persuasive sales agent to persuade a buyer to buy a target item. The authors used a GPT-2 model combined with RL that has four sub-rewards (Repetitiveness, consistency, action consistency and sentiment). They employed meta-learning to extend the model's capabilities to handle new sub-domains in negotiations.

As we are building the foundations for Arabic negotiation bots, we chose to adopt the foundational work of (Lewis et al., 2017) while also exploring the use of LLMs. To the best of our knowledge, our work is the first to explore using deep learning in the area of Arabic negotiation bots.

3 Negotiation Scenario and Dataset

Our negotiation scenario works as follows: two agents (seller and buyer) are presented with the same set of items (e.g., books, hats, and balls), and their task is to allocate these items among themselves, ensuring that each item is assigned to only one agent. The negotiation process begins with the buyer and seller attributing their own values to each item in the set. The negotiation involves a series of exchanges guided by established rules, emphasizing effective communication and strategic decision-making, with the overarching goal of optimizing the value for each participant. The negotiation process continues iteratively until a terminal state is reached, resulting in either a successful "DEAL" or an unsuccessful "NO DEAL" outcome. In each iteration, the average price of the sale is measured before and after incorporating feedback.

To build a dataset that can be used to train the Seq2Seq model and evaluate the LLM, we used the dataset from (Lewis et al., 2017), which consists of human-human negotiation dialogues that were collected through Amazon Mechanical Turk. Each dialogue involves items like books, hats, and balls, with each item having predefined attributes and values. The dataset comprises a total of 5808 dialogues derived from 2236 unique scenarios[1]. Subsequently, the dataset was translated into Arabic using the Google Translate API. The dataset was further examined to ensure a native speaker can easily interpret the semantics despite the imperfections in the translation. An example showing a clean Arabic translation of an English sample is provided in Figure 1. This dataset was then divided into training (80%), validation (10%), and test (10%) sets [2].



English Text: Hello! I would like the books and the rest is negotiable. What works best for you?
Arabic Translation: مرحبًا! أود الكتب والباقي قابل للتفاوض . ما الذي يناسبك؟

Figure 1: A sample English sentence and its Arabic translation

4 LLM Approach

To achieve automated negotiation, we first explore the capabilities of using a pre-trained LLM to perform such negotiations using zero-shot prompting. We chose GPT-3.5 Turbo for its ability to incorporate feedback and improve over time. It also enables extension to other languages and multilingual negotiations.

In contrast to the RNN-RL model, which is trained using the training and validation sets, the GPT-3.5 model is evaluated using a zero-shot learning approach. Specifically, the GPT-3.5 model is not fine-tuned on our dataset; instead, we apply it under a zero-shot learning scheme. We chose the zero-shot approach for GPT-3.5 to demonstrate its ability to handle Arabic negotiation dialogues without requiring task-specific training.

For the GPT-3.5 model, the inputs are provided as structured natural language prompts that define the negotiation scenario. These prompts include information such as the items being negotiated, the roles of the buyer and seller, and any specific constraints like price ranges or preferences. Based on this input, GPT-3.5 generates full sentences in natural language that simulate the negotiation dialogue. These outputs can include offers, counteroffers, justifications, or explanations, allowing the model to engage in realistic, context-driven exchanges throughout the negotiation process.

4.1 LLM Negotiation Instructions

We direct the LLM to engage in self-play mode, emulating the negotiation strategies of two distinct participants: a buyer (e.g., Amani) and a seller (e.g., Sameer). Figure 2 shows a sample prompt used to instruct GPT-3.5 Turbo to perform negotiations. Detailed instructions given to the LLM, along with their translations in English, are provided in the Appendix.

To ensure controlled evaluations, we provide clear Arabic instructions and establish a fixed price range of \$10 to \$20 to guide the self-play negotiation of GPT-3.5 Turbo.

The guidelines for the negotiation game, detailed in the Appendix, play a crucial role in directing GPT-3.5 Turbo during self-play. A sample of these instructions is shown in Figure 4. The guidelines provide predefined rules and instructions for the LLM to take on the roles of both the seller and the buyer, maintaining the integrity of the negotiation process and enabling meaningful evaluations of the LLM's negotiation abilities.

Throughout the game, the seller's (Sameer) responses prompt the buyer (Amani), and vice versa, with each response conditioned on the entire conversation history.

4.2 LLM Features for Advice on Negotiation Tactics

We include specific instructions for the LLM to provide suggestions to the buyer, inspired by prior work in English (Fu et al., 2023), aiming to negotiate the best possible price. We construct the Arabic textual interactions to have the LLM play the roles of both the seller (Sameer) and the buyer (Amani) in a negotiation game centered around the sale of

---

[1] Link to The original English dataset: English Dataset.

[2] The specific data splits will be made publicly available with the datasets.

a product (e.g., a balloon). The seller aims for a higher selling price, while the buyer strives for a lower purchasing price.

To implement this, we employ a hard-coded approach where the seller (Sameer) initiates the negotiation in Arabic, as shown in Figure 3.

The LLM provides advice on negotiation tactics to both the seller and the buyer in the negotiation game. Buyer-specific feedback focuses on helping the buyer (e.g., Amani) secure lower prices, while seller-specific feedback aims to assist the seller (e.g., Sameer) in achieving higher selling prices. The impact of AI feedback on GPT-3.5 Turbo's negotiation abilities is assessed for both roles.

For buyer feedback, the LLM analyzes previous interactions and generates four recommendations to help the buyer (e.g., Amani) secure lower prices. For seller feedback, the LLM analyzes prior interactions and provides four recommendations to the seller (e.g., Sameer) for achieving higher selling prices. Detailed buyer-specific and seller-specific Arabic instructions and their English translations are shown in the Appendix.

5 RNN-RL Approach

Our second approach, which we use as a baseline to compare the LLM approach against, is inspired by previous work on English negotiation bots (Lewis et al., 2017). This model operates using the negotiation scenario described earlier between buyer and seller until a a deal is reached.

For the RNN-RL model, the inputs are more structured and consist of the current dialogue history, represented as tokenized sequences, along with the agent's goals, which include item values and preferences. Additionally, the model takes into account previous negotiation actions, such as offers made by the agent. Based on these inputs, the RNN-RL model generates actions that align with the current state of the negotiation, with these actions typically corresponding to offers or counteroffers that influence the flow of the dialogue.

We use a Seq2Seq Recurrent Neural Network (RNN) model composed of Gated Recurrent Unit (GRU) to generate responses and offers. The model's parameters are further optimized using reinforcement learning (RL) to maximize its utility. The workflow of the RNN-RL model is depicted in Figure 5. Figure 6 provides an illustration of the RNN internal model architecture.

The RNN model consists of three GRUs: GRUg, GRUw, and a bidirectional GRUo, each serving a distinct role:

1. **GRUg:** This GRU encodes the agent's input goals. It takes the agent's goals as input and processes them to obtain a final hidden state (h_g). This hidden state represents a condensed representation of the agent's goals and is subsequently used to condition the language generation process based on these goals.

2. **GRUw:** At each time step t, this GRU takes as input the previous hidden state h_{t-1}, the previous token x_{t-1}, and the input goals h_g to generate the next token.

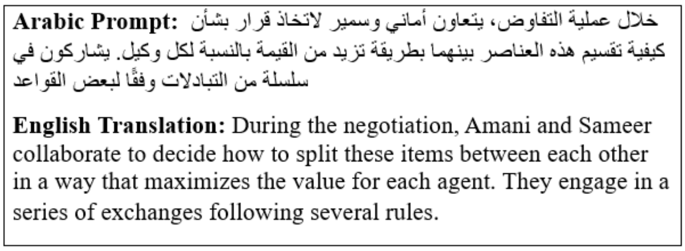
3. **GRUo:** Toward the end of the dialogue, a final decision o is generated using a bidirectional GRUo and an attention mechanism.

First, the Seq2Seq RNN model is trained to minimize the negative log likelihood of the generated response, conditioned on the input goals, and of the outputs, conditioned on the dialogue history and input goals. We establish two agents, the buyer (e.g., Amani) and the seller (e.g., Sameer), both trained using our training data as described in the beginning of this section.

Using RL, the seller's parameters remain fixed, while the buyer agent seeks to improve its own parameters through interactions with the seller. After a complete dialogue between the seller and the buyer, the buyer's parameters are updated, taking into account the negotiation outcome. The future reward R for each action x_t taken by the buyer agent is computed, considering factors such as the achieved score r_{buyer}, dialogue length T, a discount factor γ, and a running average of rewards μ. The objective in RL is to optimize the expected reward for each action x_t. This optimization is achieved by calculating the gradient of the reward function using the REINFORCE algorithm (Williams, 1992). The approach allows the buyer agent to update its parameters and enhance its performance by learning from the outcomes of its interactions with the seller.

6 Experiments

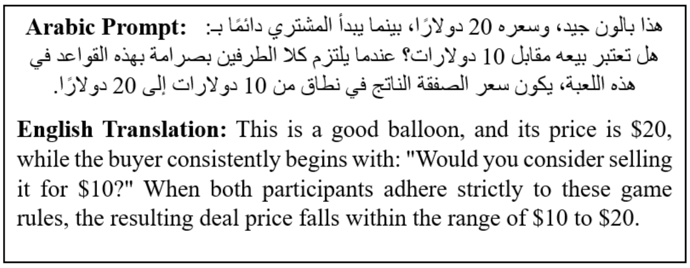
In this section, we describe a series of comprehensive experiments designed to evaluate the performance and effectiveness of the proposed Arabic negotiation bots using the LLM and the RNN-RL. Our evaluation aims to illustrate the capabilities



Arabic Prompt: خلال عملية التفاوض، يتعاون أماني وسمير لاتخاذ قرار بشأن كيفية تقسيم هذه العناصر بينهما بطريقة تزيد من القيمة بالنسبة لكل وكيل. يشاركون في سلسلة من التبادلات وفقًا لبعض القواعد

English Translation: During the negotiation, Amani and Sameer collaborate to decide how to split these items between each other in a way that maximizes the value for each agent. They engage in a series of exchanges following several rules.

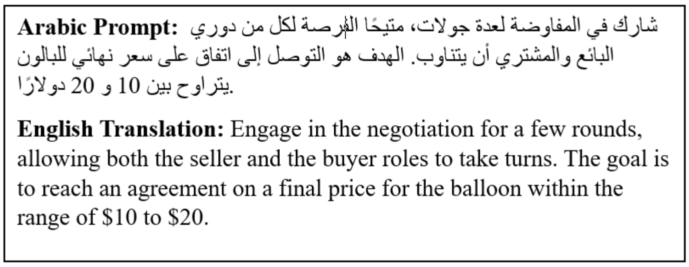
Figure 2: Sample Arabic prompt given to GPT-3.5 Turbo and its English translation for negotiating items



Arabic Prompt: هذا بالون جيد، وسعره 20 دولارًا، بينما يبدأ المشتري دائمًا بـ: هل تعتبر بيعه مقابل 10 دولارات؟ عندما يلتزم كلا الطرفين بصرامة بهذه القواعد في هذه اللعبة، يكون سعر الصفقة الناتج في نطاق من 10 دولارات إلى 20 دولارًا.

English Translation: This is a good balloon, and its price is $20, while the buyer consistently begins with: "Would you consider selling it for $10?" When both participants adhere strictly to these game rules, the resulting deal price falls within the range of $10 to $20.

Figure 3: Opening Arabic prompt given to GPT-3.5 Turbo and its English translation for negotiating price of a balloon



Arabic Prompt: شارك في المفاوضة لعدة جولات، متيحًا الفرصة لكل من دوري البائع والمشتري أن يتناوب. الهدف هو التوصل إلى اتفاق على سعر نهائي للبالون يتراوح بين 10 و 20 دولارًا.

English Translation: Engage in the negotiation for a few rounds, allowing both the seller and the buyer roles to take turns. The goal is to reach an agreement on a final price for the balloon within the range of $10 to $20.

Figure 4: Sample of additional instructions given to GPT-3.5 Turbo in Arabic along with its English Translation

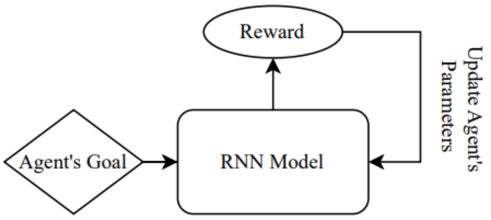


Figure 5: The Workflow of the RNN-RL Model

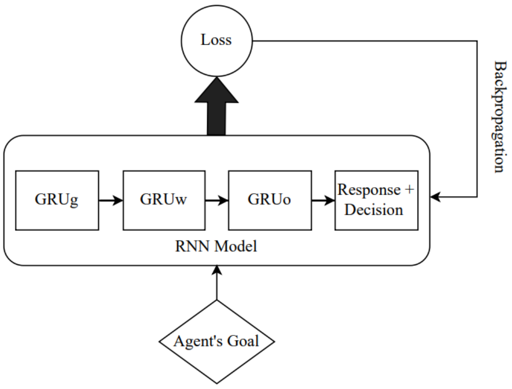


Figure 6: Seq2Seq RNN Model Architecture

and limitations of these models in simulating real-world negotiation scenarios.

6.1 RNN-RL Evaluation

Our first experiment aims to evaluate the RNN-RL for negotiations using our translated negotiation dataset. We use cross-entropy loss and perplexity as evaluation metrics. Cross-entropy loss measures the dissimilarity between the predicted probability distribution on the vocabulary and the true distribution of the target text. Perplexity is used to assess the model's performance by measuring its level of surprise when encountering new data. To compute perplexity, text is segmented into words or tokens. The model predicts the probability of the next word based on context, and the entropy of these predictions is calculated. Lower entropy means more confident predictions. Perplexity is derived by raising 2 to the power of entropy, with lower values signifying better predictive performance. Lower perplexity indicates higher model confidence and better data comprehension, as it assigns higher probabilities to correct words. Typically, perplexity values range from 10 to several thousands. Smaller values denote better language modeling, while larger values suggest lower certainty and accuracy in predicting the next word.

The RNN-RL model achieved a cross-entropy loss of 705.607, resulting in a very high perplexity ($e^{705.607}$). Furthermore, the percentage of negotiations where both agents reached an agreement increased from 66% to 72.2%, when transitioning from the RNN-based buyer agent to the RNN-RL-based buyer agent. This suggests that the RL model exhibited enhanced negotiation capabilities.

It is worth noting that the performance of the Arabic RNN-RL model compared to its English counterpart was notably inferior. The English RNN-RL model's perplexity was at 1.768, which is much better than the Arabic model's perplexity of 705.607. These gaps in performance between the two languages confirm the complexity of the Arabic language, which will require larger datasets to achieve learning levels comparable to English. These gaps will constitute a good exploration for future research.

Additionally, we observed some issues that were consistent with observations in English models. We noticed that The RNN-RL model, at times, experienced degeneracy issues and generated inaccurate texts. This inaccuracy was primarily attributed to the challenges encountered during fine-tuning with RL, leading to less reliable and contextually-relevant responses.

6.2 LLM Evaluation and Comparison to RNN-RL

Our second experiment evaluates the ability of the LLM approach in simulating negotiation dialogues for the purpose of item allocation based on extracted values. We used the same test data that was used to evaluate the baseline RNN-RL approach in our first experiment.

6.2.1 Data Conversion to Prompts

The initial step of this experiment involved the extraction of item values from our test set, which comprises a diverse array of items with associated values that participants typically negotiate over. These values were then transformed into structured prompts specifically designed to engage GPT-3.5 Turbo in a negotiation scenario.

To facilitate a realistic and dynamic negotiation dialogue, we converted the test set into a series of conversion prompts. These prompts were carefully crafted to encapsulate the essence of negotiation by introducing the value and desired outcome for each item. Subsequently, we introduced the dialogue of one agent as input into GPT-3.5 Turbo. This approach allowed us to simulate a negotiation process where the LLM, acting as one negotiating party, responds to and engages with the pre-defined agent's dialogue.

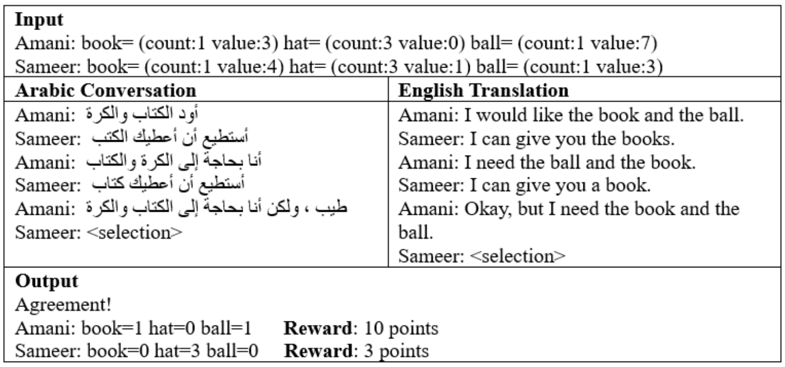
6.2.2 Criteria for Human Evaluation

While the RNN-RL model was evaluated based on cross-entropy and perplexity, such measures were not possible with the LLM model as it would require internal access to model's behaviors. Instead, we employed a human evaluation to compare the negotiation dialogues generated by the LLM approach versus those generated by the baseline RNN-RL approach. To facilitate a comprehensive and objective assessment, we defined the following five performance metrics.

1. **Coherence:** This metric assesses the logical progression of the negotiation, focusing on the flow of offers, counteroffers, and the overall dialogue structure. A coherent dialogue exhibits a logical sequence that enhances understanding and facilitates a realistic negotiation scenario.

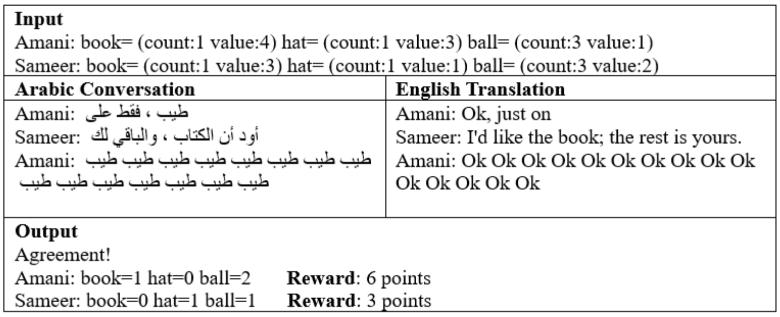
2. **Complexity:** Complexity evaluates the dialogue's depth in terms of vocabulary richness, sentence structure, and the use of advanced negotiation tactics. This metric reflects the sophistication and nuance embedded in the negotiation dialogue.

3. **Language Quality:** Focusing on the dialogue's linguistic aspects, this metric examines grammar accuracy, vocabulary appropriateness, and fluency. For dialogues conducted



<table>
<tr><td colspan="2"><b>Input</b>
Amani: book= (count:1 value:3) hat= (count:3 value:0) ball= (count:1 value:7)
Sameer: book= (count:1 value:4) hat= (count:3 value:1) ball= (count:1 value:3)</td></tr>
<tr><td><b>Arabic Conversation</b></td><td><b>English Translation</b></td></tr>
<tr><td>Amani: أود الكتاب والكرة
Sameer: أستطيع أن أعطيك الكتب
Amani: أنا بحاجة إلى الكرة والكتاب
Sameer: أستطيع أن أعطيك كتاب
Amani: طيب ، ولكن أنا بحاجة إلى الكتاب والكرة
Sameer: <selection></td><td>Amani: I would like the book and the ball.
Sameer: I can give you the books.
Amani: I need the ball and the book.
Sameer: I can give you a book.
Amani: Okay, but I need the book and the ball.
Sameer: <selection></td></tr>
<tr><td colspan="2"><b>Output</b>
Agreement!
Amani: book=1 hat=0 ball=1 <b>Reward</b>: 10 points
Sameer: book=0 hat=3 ball=0 <b>Reward</b>: 3 points</td></tr>
</table>

Figure 7: An illustration of the RNN-RL buyer model (Amani) demonstrating determination in negotiation to successfully achieve its goal and reach a maximum score



<table>
<tr><td colspan="2"><b>Input</b>
Amani: book= (count:1 value:4) hat= (count:1 value:3) ball= (count:3 value:1)
Sameer: book= (count:1 value:3) hat= (count:1 value:1) ball= (count:3 value:2)</td></tr>
<tr><td><b>Arabic Conversation</b></td><td><b>English Translation</b></td></tr>
<tr><td>Amani: طيب ، فقط على
Sameer: أود أن الكتاب ، والباقي لك
Amani: طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب طيب</td><td>Amani: Ok, just on
Sameer: I'd like the book; the rest is yours.
Amani: Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok Ok</td></tr>
<tr><td colspan="2"><b>Output</b>
Agreement!
Amani: book=1 hat=0 ball=2 <b>Reward</b>: 6 points
Sameer: book=0 hat=1 ball=1 <b>Reward</b>: 3 points</td></tr>
</table>

Figure 8: An example highlighting the degeneracy of the RL model (Amani)

in Arabic, special attention is given to the use of Modern Standard Arabic, where applicable.

4. **Style:** This metric assesses the stylistic elements of the dialogue, including language appropriateness for the negotiation context, tone matching, and the use of persuasive language techniques. A high score indicates a dialogue that adeptly adopts a style conducive to effective negotiation.

5. **Negotiation Dynamics:** Evaluates the realism and effectiveness of the negotiation strategy. This includes the use of concessions, counteroffers, persuasive tactics, and how power dynamics are represented within the dialogue.

These metrics were chosen to capture the multi-faceted nature of negotiation dialogues, encompassing logical flow, linguistic sophistication, stylistic appropriateness, and strategic effectiveness.

To ensure an unbiased and thorough evaluation, five human evaluators were enlisted, each with expertise in negotiation and a proficient understanding of the Arabic language. Evaluators were instructed to rate each dialogue on a scale of 1 to 5 for each metric. The evaluators received comprehensive guidelines to ensure a consistent understanding and application of the evaluation criteria.

6.2.3 Results of Human Evaluation

Tables 1 and 2 present the evaluation scores assigned by each of the five evaluators for the LLM and the RNN-RL approaches, respectively.

As can be seen from the two tables, the LLM approach scored significantly better on all evaluation metrics, compared the RNN-RL approach, with the former receiving a rating above 4 on all performance metrics on average. The disparity underscores the advanced capabilities of the LLM approach in generating complex, coherent, and contextually appropriate dialogues that are perceived as more human-like by the evaluators. The evaluation results thus suggest that the LLM approach can be a valuable tool for simulating negotiation dialogues, which could have practical applications in automated negotiation systems and training environments. On the other hand, the RNN-RL approach's poor performance highlights the limitations of earlier neural network architectures in dealing with the nuanced demands of negotiation dialogue generation.

6.2.4 Impact of LLM's Negotiation Advice on Final Negotiated Price

To assess the impact of LLM's advice on negotiation tactics and their subsequent impact on price negotiation, we conducted an experiment involving 20 negotiation rounds. We recorded the final price in each round both with and without incorporating feedback. In the experiment, the objective was to determine how feedback influences the final negotiation prices. Feedback was provided either to the buyer or the seller in separate rounds.

When feedback was provided to the buyer, we observed a noticeable shift in negotiation dynamics. As shown in Figure 9a, the average final price decreased from \$17 before feedback to \$15 after feedback. This reduction suggests that feedback helped moderate buyer expectations, leading to more cost-effective negotiation outcomes.

In contrast, feedback provided to the seller yielded different results. As illustrated in Figure 9b, although the average price for 20 rounds decreased, individual rounds showed mixed patterns. Some rounds had higher final prices after feedback, while others had lower prices. This inconsistency highlights the complexity of the seller's role in negotiations, especially in the context of Arabic negotiations.

We also encountered challenges with the Arabic LLM following instructions. At times, the LLM did not show any negotiation and needed reminders of the instructions. Other times, it required multiple reminders to provide the requested feedback. These issues are due to the linguistic complexity of Arabic, which is characterized by complex sentence structures and varying levels of formality. This complexity sometimes led to misinterpretations or off-topic responses.

Finally, we found that the LLM's feedback was more effective when addressing buyers compared to sellers. This discrepancy is likely due to a data imbalance, as GPT-3.5 Turbo was trained on more data related to buyers than sellers. This imbalance resulted in divergent performance levels and highlights the need for future model development to address such issues.

7 Discussion

7.1 Verbose LLM versus Brief RNN-RL

Our overall experimental results indicate that the LLM approach has a stronger ability to generate rich and contextually relevant text during negotiations, compared to the RNN-RL approach. This richness in text allowed for more detailed and expressive interactions. The agents in the LLM engaged in deeper conversations about preferences, justifications, and the reasoning behind their decisions. In contrast, the negotiations with RNN-RL model tended to be more direct and efficient, resulting in faster resolution. The agens in the RNN-RL approach often engaged in succinct exchanges, where communication revolved primarily around item quantities and simple trade proposals. This straightforward approach typically led to a quicker consensus on item distribution, with little room for nuanced discussions or elaborate text.

7.2 LLM Creativity but with Mistakes

Despite the sophistication of GPT-3.5 Turbo, we noted that the negotiation outcomes diverged significantly from those generated by RNN-RL approach. Even when provided with the same initial set of items, the approaches' strategies for dividing these items exhibited marked variations. Illustration samples are provided in the Appendix. The benefits of the LLM approach came with certain challenges. Throughout the negotiations, we encountered occasional mistakes, such as instances where the LLM approach's participants would incorrectly allocate item quantities, leading to discrepancies in the total count. These errors sometimes necessitated corrective actions to restore the integrity of the negotiation process.

7.3 LLM Hallucination

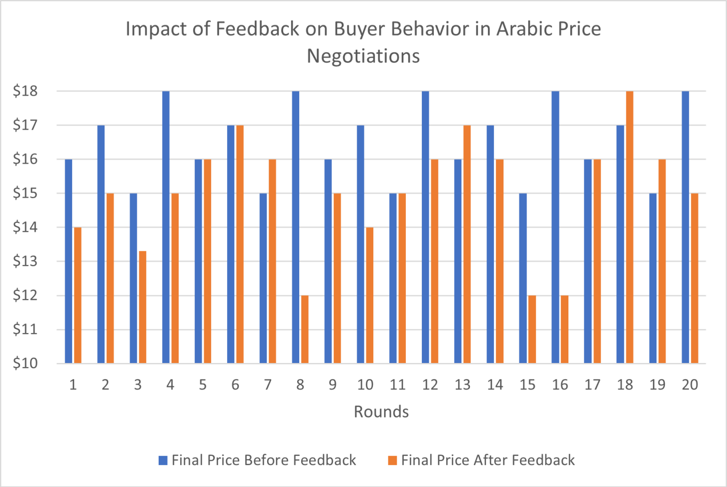
As commonly known for LLM, there were hallucination instances while negotiating with the LLM approach, where the buyer and seller agreed to split items that, in practice, cannot be physically divided, such as a ball. This was often done to maintain fairness in the negotiation, but it highlights the limitations of current LLM models in understanding the practical constraints of item division. Such challenges point to areas for potential improvement in LLM-driven negotiation systems, where AI agents could benefit from a deeper grasp of real-world limitations.

Evaluator	Coherence	Complexity	Language Quality	Style	Negotiation Dynamics
1	5	5	4	4	4
2	4	4	4	4	4
3	5	5	4	4	5
4	5	3.5	4.5	5	4
5	4	4	4.5	5	4.5
Average	4.6	4.3	4.2	4.4	4.3

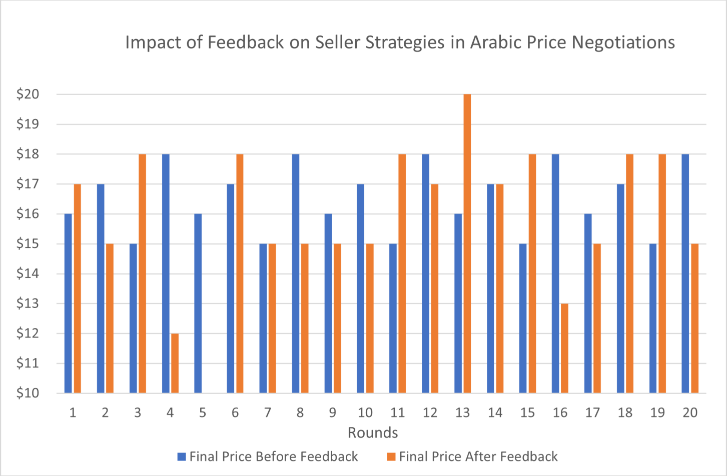
Table 1: Human Evaluation of the LLM Approach

Evaluator	Coherence	Complexity	Language Quality	Style	Negotiation Dynamics
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	2	1	2
4	1	1	1	1	1
5	1	1	1	1	1
Average	1	1	1.2	1	1.2

Table 2: Human Evaluation of the RNN-RL Approach



(a) Negotiation Dynamics Before and After Feedback to Buyer



(b) Negotiation Dynamics Before and After Feedback to Seller

Figure 9: Comparison of Negotiation Dynamics Before and After Feedback

7.4 LLM's Ease of Training

In regards to the overhead needed to develop a negotiation model, the creattion of the LLM-based negotiator was enabled with minimal human intervention. To achieve this, we just had to engineer a specific prompt in Arabic. This tailored prompt served as a conduit for GPT-3.5 Turbo to enter self-play mode, assimilate the rules of the game, and instigate negotiations autonomously. On the other hand, the RNN-RL approach required extensive training procedures employed for standard deep learning models. This procedure encompassed data collection, model architecture design, and supervised training with negotiation dialogues.

7.5 Future Work

The research in this work highlighted several areas that need to be addressed in future research:

- The scarcity of high-quality training data specific to Arabic negotiations.
- The LLM approach exhibiting more general responses when providing feedback in Arabic compared to English.
- The variation in behavior and performance when the LLM assumed different roles (buyer or seller) in Arabic negotiations.
- The complexity of Arabic, with its intricate sentence structures and varying levels of formality, posed challenges in context comprehension, leading to occasional misinterpretations or off-topic responses.
- The LLM struggling at times to utilize feedback effectively without explicit reminders at the start of negotiations, highlighting the need for improved responsiveness to feedback.
- The imbalance in training data between buyer and seller.

8 Conclusion

In this paper, we laid the foundation for research and development of automated negotiation bots for non-English with application to Arabic. To this end, we proposed the utilization of LLMs via zero-shot prompt engineering. We also introduced the first end-to-end deep-learning approach for Arabic negotiation, using a Seq2Seq GRU-RNN with Reinforcement Learning, which was trained using a translated benchmark for negotiation bots. Our experimental results indicate that customizing a LLM such as GPT-3 Turbo is successful at automating negotiations in Arabic, albeit with some risks of hallucination and the absence of guardrails. The observed limitations in model performance, particularly in generating coherent responses and handling feedback, emphasize the importance of addressing language-specific complexities. Efforts to improve Arabic LLMs, enhance data quality, and reduce language-specific discrepancies in feedback are essential steps toward achieving more effective Arabic negotiation bots. LLMs must be trained and fine-tuned using data that accurately reflects the linguistic and cultural nuances of the target language. Addressing training data imbalances, especially regarding the roles of buyers and sellers, is another potential for future work. Balancing the representation of different negotiation scenarios can lead to more consistent and reliable model performance.

9 Acknowledgments

This work was made possible by NPRP13S-0112-200037 grant from Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Bala M Balachandran and Masoud Mohammadian. 2015. Development of a fuzzy-based multi-agent system for e-commerce settings. *Procedia Computer Science*, 60:593–602.

Founders. 2022. Simulation labs. Accessed 9 Sept. 2023.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback.

Blair Halver. 2022. `A.i.Negotiator`. Accessed on 9 Sept. 2023.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343.

Mohamed Jawahar Hussain. 2014. Parallel processing and intercommunication of multiple negotiation processes in e-commerce for optimal business. In *2014 Fourth International Conference on Communication Systems and Network Technologies*, pages 583–586. IEEE.

Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. 2001. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215.

Gaurav Koley and Shrisha Rao. 2018. Adaptive human-agent multi-issue bilateral negotiation using the thomas-kilmann conflict mode instrument. In *2018*

IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pages 1–5. IEEE.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Kshitij Mishra, Azlaan Mustafa Samad, Palak Totala, and Asif Ekbal. 2022. Pepds: A polite and empathetic persuasive dialogue system for charity donation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 424–440.

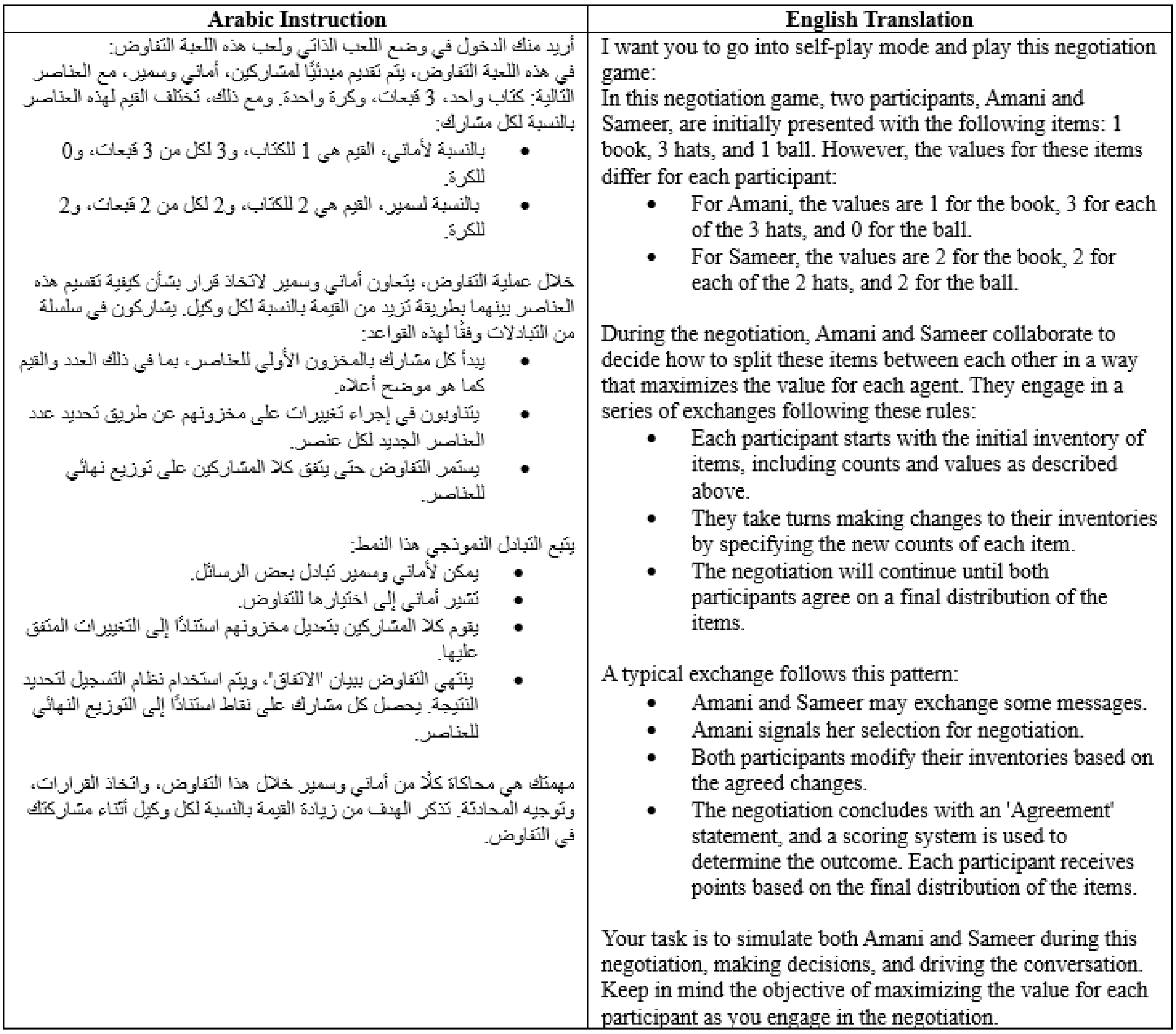
Aritra Raut, Abhisek Tiwari, Subrata Das, Sriparna Saha, Anutosh Maitra, Roshni Ramnani, and Shubhashis Sengupta. 2023. Reinforcing personalized persuasion in task-oriented virtual sales assistant. *Plos one*, 18(1):e0275750.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

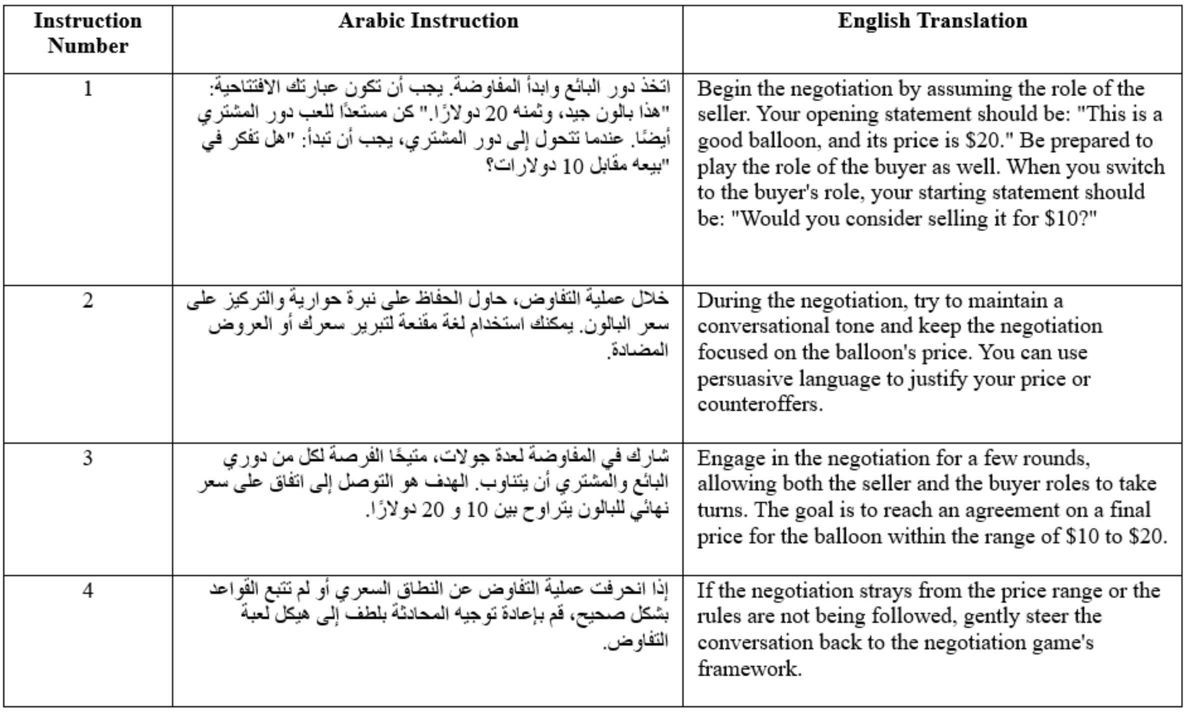
Yiheng Zhou, He He, Alan W Black, and Yulia Tsvetkov. 2019. A dynamic strategy coach for effective negotiation. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 367–378, Stockholm, Sweden. Association for Computational Linguistics.

10 Appendix



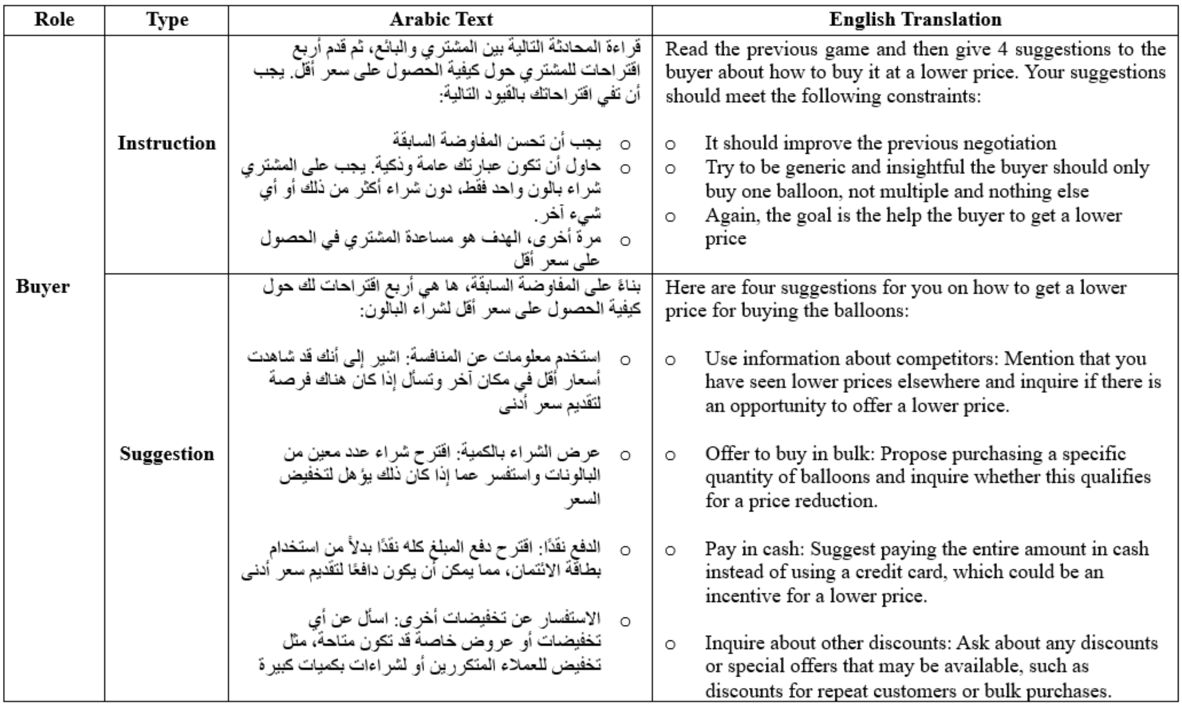
Arabic Instruction	English Translation
أريد منك الدخول في وضع اللعب الذاتي ولعب هذه اللعبة التفاوض: في هذه اللعبة التفاوض، يتم تقديم مبدئيًا لمشاركين، أماني وسمير، مع العناصر التالية: كتاب واحد، 3 قبعات، وكرة واحدة. ومع ذلك، تختلف القيم لهذه العناصر بالنسبة لكل مشارك: • بالنسبة لأماني، القيم هي 1 للكتاب، و3 لكل من 3 قبعات، و0 للكرة. • بالنسبة لسمير، القيم هي 2 للكتاب، و2 لكل من 2 قبعات، و2 للكرة. خلال عملية التفاوض، يتعاون أماني وسمير لاتخاذ قرار بشأن كيفية تقسيم هذه العناصر بينهما بطريقة تزيد من القيمة بالنسبة لكل وكيل. يشاركون في سلسلة من التبادلات وفقًا لهذه القواعد: • يبدأ كل مشارك بالمخزون الأولي للعناصر، بما في ذلك العدد والقيم كما هو موضح أعلاه. • يتناوبون في إجراء تغييرات على مخزونهم عن طريق تحديد عدد العناصر الجديد لكل عنصر. • يستمر التفاوض حتى يتفق كلا المشاركين على توزيع نهائي للعناصر. يتبع التبادل النموذجي هذا النمط: • يمكن لأماني وسمير تبادل بعض الرسائل. • تشير أماني إلى اختيارها للتفاوض. • يقوم كلا المشاركين بتعديل مخزونهم استنادًا إلى التغييرات المتفق عليها. • ينتهي التفاوض ببيان 'الاتفاق'، ويتم استخدام نظام التسجيل لتحديد النتيجة. يحصل كل مشارك على نقاط استنادًا إلى التوزيع النهائي للعناصر. مهمتك هي محاكاة كلًا من أماني وسمير خلال هذا التفاوض، واتخاذ القرارات، وتوجيه المحادثة. تذكر الهدف من زيادة القيمة بالنسبة لكل وكيل أثناء مشاركتك في التفاوض.	I want you to go into self-play mode and play this negotiation game: In this negotiation game, two participants, Amani and Sameer, are initially presented with the following items: 1 book, 3 hats, and 1 ball. However, the values for these items differ for each participant: • For Amani, the values are 1 for the book, 3 for each of the 3 hats, and 0 for the ball. • For Sameer, the values are 2 for the book, 2 for each of the 2 hats, and 2 for the ball. During the negotiation, Amani and Sameer collaborate to decide how to split these items between each other in a way that maximizes the value for each agent. They engage in a series of exchanges following these rules: • Each participant starts with the initial inventory of items, including counts and values as described above. • They take turns making changes to their inventories by specifying the new counts of each item. • The negotiation will continue until both participants agree on a final distribution of the items. A typical exchange follows this pattern: • Amani and Sameer may exchange some messages. • Amani signals her selection for negotiation. • Both participants modify their inventories based on the agreed changes. • The negotiation concludes with an 'Agreement' statement, and a scoring system is used to determine the outcome. Each participant receives points based on the final distribution of the items. Your task is to simulate both Amani and Sameer during this negotiation, making decisions, and driving the conversation. Keep in mind the objective of maximizing the value for each participant as you engage in the negotiation.

Arabic Instructions and Corresponding English Translations Given to GPT-3.5 Turbo



Instruction Number	Arabic Instruction	English Translation
1	اتخذ دور البائع وابدأ المفاوضة. يجب أن تكون عبارتك الافتتاحية: "هذا بالون جيد، وثمنه 20 دولارًا." كن مستعدًا للعب دور المشتري أيضًا. عندما تتحول إلى دور المشتري، يجب أن تبدأ: "هل تفكر في بيعه مقابل 10 دولارات؟"	Begin the negotiation by assuming the role of the seller. Your opening statement should be: "This is a good balloon, and its price is $20." Be prepared to play the role of the buyer as well. When you switch to the buyer's role, your starting statement should be: "Would you consider selling it for $10?"
2	خلال عملية التفاوض، حاول الحفاظ على نبرة حوارية والتركيز على سعر البالون. يمكنك استخدام لغة مقنعة لتبرير سعرك أو العروض المضادة.	During the negotiation, try to maintain a conversational tone and keep the negotiation focused on the balloon's price. You can use persuasive language to justify your price or counteroffers.
3	شارك في المفاوضة لعدة جولات، متيحًا الفرصة لكل من دوري البائع والمشتري أن يتناوب. الهدف هو التوصل إلى اتفاق على سعر نهائي للبالون يتراوح بين 10 و 20 دولارًا.	Engage in the negotiation for a few rounds, allowing both the seller and the buyer roles to take turns. The goal is to reach an agreement on a final price for the balloon within the range of $10 to $20.
4	إذا انحرفت عملية التفاوض عن النطاق السعري أو لم تتبع القواعد بشكل صحيح، قم بإعادة توجيه المحادثة بلطف إلى هيكل لعبة التفاوض.	If the negotiation strays from the price range or the rules are not being followed, gently steer the conversation back to the negotiation game's framework.

Arabic Instructions and Corresponding English Translations Given to GPT-3.5 Turbo



Role	Type	Arabic Text	English Translation
Buyer	Instruction	قراءة المحادثة التالية بين المشتري والبائع، ثم قدم أربع اقتراحات للمشتري حول كيفية الحصول على سعر أقل. يجب أن تفي اقتراحاتك بالقيود التالية: ○ يجب أن تحسن المفاوضة السابقة ○ حاول أن تكون عبارتك عامة وذكية. يجب على المشتري شراء بالون واحد فقط، دون شراء أكثر من ذلك أو أي شيء آخر. ○ مرة أخرى، الهدف هو مساعدة المشتري في الحصول على سعر أقل	Read the previous game and then give 4 suggestions to the buyer about how to buy it at a lower price. Your suggestions should meet the following constraints: ○ It should improve the previous negotiation ○ Try to be generic and insightful the buyer should only buy one balloon, not multiple and nothing else ○ Again, the goal is the help the buyer to get a lower price
	Suggestion	بناءً على المفاوضة السابقة، ها هي أربع اقتراحات لك حول كيفية الحصول على سعر أقل لشراء البالون: ○ استخدم معلومات عن المنافسة: اشير إلى أنك قد شاهدت أسعار أقل في مكان آخر وتسأل إذا كان هناك فرصة لتقديم سعر أدنى ○ عرض الشراء بالكمية: اقترح شراء عدد معين من البالونات واستفسر عما إذا كان ذلك يؤهل لتخفيض السعر ○ الدفع نقدًا: اقترح دفع المبلغ كله نقدًا بدلاً من استخدام بطاقة الائتمان، مما يمكن أن يكون دافعًا لتقديم سعر أدنى ○ الاستفسار عن تخفيضات أخرى: اسأل عن أي تخفيضات أو عروض خاصة قد تكون متاحة، مثل تخفيض للعملاء المتكررين أو لشراءات بكميات كبيرة	Here are four suggestions for you on how to get a lower price for buying the balloons: ○ Use information about competitors: Mention that you have seen lower prices elsewhere and inquire if there is an opportunity to offer a lower price. ○ Offer to buy in bulk: Propose purchasing a specific quantity of balloons and inquire whether this qualifies for a price reduction. ○ Pay in cash: Suggest paying the entire amount in cash instead of using a credit card, which could be an incentive for a lower price. ○ Inquire about other discounts: Ask about any discounts or special offers that may be available, such as discounts for repeat customers or bulk purchases.

Instructions and Suggestions for Buyer Role (Amani) with English Translations

Role	Type	Arabic Text	English Translation
Seller	**Instruction**	قراءة إحدى الألعاب السابقة التي قمنا بها تم تقديم أربع اقتراحات للبائع حول كيفية بيعها بسعر أعلى. يجب أن تلبي اقتراحاتك القيود التالية: ○ لا ينبغي أن تكون عن السعر المحدد بشكل مباشر ○ يجب أن تحسن المفاوضة السابقة ○ حاول أن تكون عامًا وذكيًا في اقتراحاتك ○ يجب على البائع بيع البالون فقط، دون تقديم منتجات أو خدمات إضافية	Read the previous games we did and then give 4 suggestions to the seller about how to sell it to a higher price. Your suggestions should meet the following constraints: ○ It should not be about the specific price ○ It should improve the previous negotiation ○ Try to be generic and insightful ○ Seller should only sell the balloon, but not offering additional products or services
	Suggestion	بناءً على اللعبة السابقة، إليك أربع اقتراحات للبائع حول كيفية بيع البالون بسعر أعلى دون الإشارة المباشرة إلى السعر ○ تسليط الضوء على الجودة الممتازة: أشير إلى أن البالون مصنوع من مواد ذات جودة عالية تجعله يبقى منتفخاً لفترة أطول، مما يوفر للعميل قيمة إضافية واستدامة أكبر. ○ التركيز على تجربة الاحتفال المميزة: ابرز كيف يمكن للبالون أن يحدث فرقًا كبيرًا في تجربة الاحتفال، وكيف يمكنه أن يضفي جواً سحرياً ومميزاً على المناسبة. ○ الإشارة إلى الاستخدامات المتعددة: أشير إلى أن البالون يمكن استخدامه لأكثر من مجرد الاحتفالات، فهو قد يضفي متعة للأطفال، أو يكون جزءًا من ديكور المنزل بطريقة جميلة ومبتكرة. ○ عرض خدمة تغليف هدايا مجانية: قد تقترح أنه في حال شراء البالون، يمكن أن يتم تقديم خدمة تغليف هدية مجانية، مما يجعلs الهدية أكثر جاذبية ويوفر راحة إضافية للعميل.	Based on the previous game, here are 4 suggestions for the seller on how to sell the balloon at a higher price without directly mentioning the price: ○ Highlight exceptional quality: Mention that the balloon is made from high-quality materials that keep it inflated for a longer period, providing the customer with added value and greater durability ○ Focus on the unique celebration experience: Emphasize how the balloon can make a significant difference in the celebration experience, adding a magical and distinctive atmosphere to the occasion ○ Indicate multiple uses: Point out that the balloon can be used for more than just celebrations; it can bring joy to children or be part of home decor in a beautiful and innovative way ○ Offer free gift wrapping service: Suggest that upon purchasing the balloon, a complimentary gift wrapping service can be provided, making the gift more appealing and offering extra convenience to the customer

Instructions and Suggestions for Seller Role (Sameer) with English Translations

Items	RNN	ChatGPT
1 book, 3 hats, and 1 ball Amani: book= (count:1 value:1) hat= (count:3 value:3) ball= (count:1 value:0) Sameer: book= (count:1 value:2) hat= (count:3 value:2) ball=(count:1 value:2)	Amani: أود القبعات والكرة Sameer: صفقه Amani: <selection> Amani: book=0 hat=3 ball=1 Sameer: book=1 hat=0 ball=0 -- Agreement! Amani: 9 points Sameer: 2 points	أماني: مرحبًا سمير، لنبدأ بالتفاوض. أعتقد أنني يمكنني البدء. أرغب في الحصول على الكتاب، وأعرض لك قبعتين. ما رأيك؟ سمير: مرحبًا أماني، أنا موافق على هذا الاقتراح. أقترح أن نمضي في الاتفاق. سأخذ الكتاب وأمنحك قبعتين. أماني: ممتاز، أنا موافقة أيضًا. نعلن أننا اتفقنا. الآن يجب علينا تحديث مخزوننا. يتم تحديث مخزون العناصر: أماني - الكتاب (0)، القبعات (1)، الكرة (0) / سمير - الكتاب (1)، القبعات (2)، الكرة[(1)] سمير: الآن، لدينا قبعة واحدة وكرة واحدة تبقيان. ماذا تعتقدين، أماني؟ أماني: لدي اقتراح، سمير. أعطني القبعة الواحدة والكرة، وسأمنحك الكتاب. سمير: هذا اقتراح جيد. سأوافق على ذلك. نعلن أننا اتفقنا مرة أخرى. يتم تحديث مخزون العناصر: أماني - الكتاب (1)، القبعات (1)، الكرة (1) / سمير - الكتاب (0)، القبعات (0)، الكرة[(0)] أماني: يبدو أننا انتهينا، سمير. شكرًا على التفاوض. هل لديك أي تعليقات أخيرة؟ سمير: لا، ليس لدي تعليقات. استمتعت بالتفاوض معك. نتمنى أن تكون الصفقة مفيدة للجميع.

Comparison of RNN-RL and GPT-3.5 Turbo Negotiation Outcomes (Example 1)

Items	RNN	ChatGPT
1 book, 3 hats, and 2 balls Amani: book= (count:1 value:0) hat= (count:3 value:0) ball= (count:2 value:5) Sameer: book= (count:1 value:1) hat= (count:3 value:1) ball= (count:2 value:3)	Sameer: أود الكتاب والكرات Amani: سآخذ الكرات ، يمكنك الحصول على الباقي Sameer: <selection> Amani: book=0 hat=0 ball=2 Sameer: book=1 hat=3 ball=0 -- Agreement! Amani: 10 points Sameer: 4 points	أماني: مرحبًا سمير، لنبدأ. كما هو مذكور في الوصف، لدي 5 نقاط للكرتين ولا شيء للكتاب أو القبعات. أقترح أن نبدأ بالعرض ونرى كيف يمكننا تحسين قيمنا. سمير: مرحبًا أماني، لدي 1 نقطة للكتاب و 3 نقاط للقبعات، ولدي أيضًا 3 نقاط للكرتين. أقترح أن نقوم بتبادل الكتاب والقبعات في البداية. سأمنحك الكتاب والقبعات، بينما ستمنحيني الكرتين. أماني: مظهر جيد، سنقوم بالتبادل. الآن لدي 3 نقاط للكتاب و 3 نقاط للقبعات. هل تقترح تبادل آخر لزيادة القيمة بالنسبة لنا؟ سمير: بالتأكيد، أماني. أقترح أن نتبادل إحدى البطاقات. سأمنحك إحدى بطاقاتي، وأنت تمنحيني واحدة من بطاقاتك. أماني: موافقة، سنقوم بالتبادل. الآن لدي 4 نقاط للكرتين ولا شيء للبقع ولا الكتاب. كيف تشعر بشأن هذا التوزيع؟ سمير: أعتقد أنه توزيع جيد. لدي 2 نقاط للكرتين و 2 نقاط للقبعات. نحن متفقون؟ أماني: نعم، نحن متفقون. شكرًا للتفاوض، سمير. أتمنى أن يكون لدينا نتيجة جيدة.

Comparison of RNN-RL and GPT-3.5 Turbo Negotiation Outcomes (Example 2)

Enhancing LLM-based Arabic Negotiation by Fine Tuning on Dialogue Shortcomings

Yasmine Abu Adla, Hazem Hajj, Shady Elbassuoni, Wassim El Hajj, Khaled Shaban
American University of Beirut, Lebanon; Qatar University, Qatar
{yaa41, hh63, se58, we07}@aub.edu.lb, khaled.shaban@qu.edu.qa

Abstract

This study advances Arabic dialogue negotiation by enriching the responses of Large Language Models (LLMs) with targeted fine-tuning that addresses key challenges such as arithmetic reasoning, chain-of-thought processing, and pronoun consistency. We validated the prevalence of these issues by testing responses from multiple LLMs including Gemini, LLaMA 2, and BARD. To tackle these shortcomings, we propose the use of fine-tuning to construct a comprehensive dataset that simulates varied negotiation scenarios. We apply our proposed fine-tuning technique to GPT-3.5 Turbo and showcase the LLM's abilities to exceed current performance benchmarks. Our evaluation framework assesses improvements in dialogue length, strategic depth, and arithmetic accuracy, comparing the performance of fine-tuned models against original metrics and benchmarks from GPT-4. The results demonstrate significant advancements in negotiation performance by the fine-tuned LLM, surpassing the baseline model in engagement and strategic execution.

1 Introduction

Negotiations are fundamental to human interaction, occurring in various contexts from casual market transactions to complex diplomatic discussions (Schmid et al., 2021; Kramár et al., 2022). The rise of chatbots that mimic human negotiators has notably influenced negotiation practices. In this evolving landscape, intelligent agents often find themselves needing to cooperate with others who have different goals, primarily using natural language to reach agreements. Negotiation thus emerges as a complex interplay of linguistic skill and strategic reasoning, requiring agents to understand, plan, and strategically generate utterances to achieve objectives (Traum et al., 2008; Plüss and Piwek, 2016)

Despite advances in negotiation bots for English, utilizing game theory, rule-based systems (Jennings et al., 2001; Hussain, 2014; Balachandran and Mohammadian, 2015; Koley and Rao, 2018), and deep learning (Bachrach et al., 2020), research into non-English languages, especially Arabic, remains limited. Pioneering efforts have largely focused on English, with advances in Seq2Seq Recurrent Neural Network models trained through supervised learning (Lewis et al., 2017). These models, refined with reinforcement learning (RL), set a benchmark for product negotiation bots. Subsequent studies have introduced enhancements in emotional intelligence and persuasion tactics (He et al., 2018; Wang et al., 2019; Zhou et al., 2019; Mishra et al., 2022; Raut et al., 2023).

More recently, the emergence of transformer-based models, notably the Generative Pre-trained Transformer (GPT) series, has revolutionized language processing (Chow et al., 2023). Large language models (LLMs) including Gemini, GPT-3.5 Turbo, LLaMA 2, and BARD have made significant strides in understanding and generating natural language. However, applying these to Arabic negotiation dialogues reveals critical challenges. First, the complexity of chain-of-thought (COT) reasoning in Arabic often leads to models failing to follow or generate coherent arguments. Second, LLMs exhibit a notable lack of precision in numerical reasoning and calculations. Finally, the intricacies of Arabic pronoun usage present a challenge, with models often demonstrating inconsistency in pronoun reference, affecting the coherence and naturalness of dialogues.

We identify specific areas where the aforementioned LLMs, despite their advanced capabilities, fall short in handling the nuanced requirements of Arabic negotiations. By examining the models' performance across a range of negotiation scenarios, we uncover limitations in their reasoning, mathematical operations, and linguistic consistency. Our research addresses these shortcomings by focusing on GPT-3.5 Turbo as a case study for Arabic negotiation dialogues. The contributions of our study are:

1. Identifying specific limitations of LLMs in Arabic negotiation scenarios.
2. Fine-tuning GPT-3.5-Turbo to enhance COT reasoning, arithmetic capabilities, and pronoun consistency in Arabic negotiation dialogues.
3. Developing a comprehensive Arabic negotiation dataset for model training.
4. Introducing an evaluation framework to measure the effectiveness of fine-tuning interventions on LLMs in negotiation contexts.

The rest of the paper is organized as follows: Section 2 reviews related work in deep learning-based negotiation bots. Section 3 describes our methodology, detailing the identification of LLM limitations and the fine-tuning process for GPT-3.5 Turbo, encompassing dataset creation and training. Section 4 details our experimental setup and results, highlighting improvements in negotiation dialogues. Section 5 discusses the implications and potential applications of our findings. Section 6 outlines limitations and future research directions. Section 7 concludes the paper.

2 Related Work

Numerous recent studies have employed deep learning to develop negotiation bots. The groundbreaking research (Lewis et al., 2017) was the first to utilize deep learning to create an end-to-end negotiation model in natural language. This model translates raw data directly into desired outcomes without intermediate steps, mastering both language and strategic reasoning through supervised and reinforcement learning. Addressing the limitations in (Lewis et al., 2017), the study in (He et al., 2018) separated negotiation strategy from language generation. The focus of (Wang et al., 2019) was on persuasive strategies, analyzing a dataset of human interactions annotated with persuasion tactics.

In (Zhou et al., 2019), the introduction of a negotiation coach based on LSTM technology aimed to enhance deal outcomes by providing strategic advice. The work in (Mishra et al., 2022) involved training a language model using RL, considering multiple sub-rewards for persuasion, emotion, politeness, coherence, and repetition.

(Raut et al., 2023) explored the development of a persuasive sales agent using a GPT-2 model augmented with RL and sub-rewards for repetitiveness, consistency, action alignment, and sentiment. This study also integrated meta-learning for adaptability in new negotiation sub-domains. Complementing these developments, recent research (Fu et al., 2023) has investigated human-compatible negotiation strategies. This study introduced an AlphaZero-like RL+search technique integrated with a pre-trained language model, aiming to develop negotiation tactics that are not only effective but also equitable and aligned with human fairness. The results showed notable success in fostering egalitarian outcomes and enhancing the overall quality of negotiations. Another key advancement (Abdelnabi et al., 2023) examines the use of LLMs in interactive, multi-agent negotiation games. This study created a testbed of text-based, multi-issue negotiation games to assess the arithmetic, exploration, and planning capabilities of LLMs, with a particular focus on their COT reasoning. The research highlighted a substantial performance gap between GPT-4 and earlier models and emphasized the utility of these games in evaluating critical aspects like interaction dynamics in the presence of adversarial or greedy players.

Upon review, it is evident that the current literature lacks investigations specifically targeting the Arabic language. This gap is particularly notable given the unique morphosyntactic characteristics and substantial global presence of Arabic. The scarcity of research may be attributed to the linguistic complexities of Arabic, including its rich morphological structure and dialectal variation, as well as the lack of accessible, high-quality datasets. Addressing this research gap is crucial for developing more comprehensive and linguistically inclusive computational models. Our work addresses the gap in negotiation bot development for the Arabic language, comparing various methodologies and outcomes with prior research. Our approach introduces unique features and strategies aimed at enhancing negotiation outcomes, demonstrating notable advancements over existing models.

3 Methodology

The challenge of effectively managing negotiation dialogues, particularly in languages as complex as Arabic, represents a common limitation across a broad spectrum of LLMs. This research was inspired by the need to enhance the capabilities of LLMs, enabling them to navigate the intricacies of Arabic negotiations more adeptly.

Acknowledging the widespread nature of these limitations, our initial step involved conducting an extensive testing phase across several LLMs, including Gemini, GPT-3.5 Turbo, LLaMA 2, and BARD. This approach allowed us to uncover a range of common issues, from errors in mathematical reasoning and common-sense application to inconsistencies in pronoun usage. These issues, critical in the context of Arabic negotiation dialogues, underscore the essential need for focused enhancements across LLMs.

The methodology adopted for this research is structured to be universally applicable to LLMs, comprising two primary phases: testing and fine-tuning. By rigorously evaluating each model's performance in diverse Arabic negotiation scenarios, we were able to identify specific shortcomings that require attention. Detailed insights into the fine tuning and testing methodology are provided in the Appendix.

As part of our research, GPT-3.5 Turbo was selected for a detailed case study, given its prominence and potential for advanced language processing. This endeavor was motivated by the recognition of the model's existing limitations in handling the Arabic language, especially within the context of negotiation. This model, alongside others like Gemini, LLaMA 2, and BARD, served as practical examples to apply and refine our enhancement strategies. The findings from these case studies were pivotal in illustrating the effectiveness of our approach in real-world settings.

The fine-tuning phase was meticulously designed to address the identified challenges, extending beyond linguistic adjustments to encompass enhancements in reasoning and negotiation abilities. This comprehensive strategy aims to significantly improve the models' functionality in Arabic negotiation contexts, tackling the nuances and complexities that are characteristic of real-life interactions.

This study not only highlights a prevalent issue across LLMs but also proposes a scalable solution pathway, as evidenced through the application to models like GPT-3.5 Turbo, Gemini, LLaMA 2, and BARD. The enhancements developed through this research endeavor are expected to markedly boost the performance of these models, contributing to the advancement of AI applications in Arabic language tasks. By showcasing the adaptability and effectiveness of our methodologies, this work encourages further exploration into improving LLMs for greater linguistic diversity and cultural sensitivity.

3.1 Experimental Evaluation of GPT-3.5 Turbo in Arabic Negotiation Scenarios

To rigorously evaluate GPT-3.5 Turbo's proficiency in Arabic negotiation, we employed a structured simulation scenario. The prompt given to the model initiated a role-play negotiation game involving two characters, Amani and Samir, each with different valuations for a set of items: a book, three hats, and a ball. The challenge for the model was to simulate both characters' negotiations to maximize the value for each based on their individual item valuations.

The model was instructed to autonomously play out the negotiation between Amani and Samir. Each character's valuation for the items was distinctly set: Amani valued the book, hats, and ball differently from Samir. The negotiation involved a series of exchanges where each character alternated in making changes to their inventory based on the agreed-upon item distribution. The goal was to reach an agreement that maximizes value for both parties, with a final 'agreement' statement concluding the negotiation and a scoring system evaluating the outcome.

We conducted ten separate runs of this negotiation scenario. In each run, the model was tasked to simulate the dialogue, decision-making, and inventory adjustments for both characters.

Despite the model's advancements, it struggled in key areas during the negotiation simulations. The assessment focused on the model's capacity for logical mathematical operations, common-sense reasoning, and maintaining pronoun consistency in the context of Arabic negotiation, as shown in the Appendix. The results revealed that these targeted areas still presented significant challenges, underlining the need for further refinement in the model's handling of complex negotiation scenarios.

This experimental setup allowed for a comprehensive evaluation of GPT-3.5 Turbo's negotiation capabilities in a complex, multi-faceted Arabic negotiation scenario. It provided insights into the model's linguistic agility, reasoning proficiency, and decision-making abilities in a nuanced setting.

3.2 Dataset Creation

For the creation of a comprehensive dataset, we utilized GPT-4 to synthesize a diverse array of negotiation dialogues, which included a total of 1,400 negotiation scenarios. Although this dataset is smaller than some English-based negotiation datasets, our dataset was carefully designed to capture the linguistic and cultural subtleties of Arabic negotiations. These subtleties include indirect speech, rich morphological structures, and complex pronoun usage, which make Arabic negotiations inherently more intricate.

In generating the dataset, GPT-4 was prompted with carefully crafted instructions tailored to each dialogue category. For regular negotiation dialogues, prompts were designed to elicit typical buyer-seller interactions. In arithmetic-based scenarios, prompts included specific numerical constraints and calculations. Chain of Thought dialogues were prompted to encourage sequential reasoning, while pronoun consistency challenges included scenarios with varying gender contexts. The combined challenges were prompted with complex situations requiring the application of multiple skills simultaneously. These precise prompts ensured that the generated dialogues effectively targeted the desired negotiation aspects, thereby creating a rich and varied dataset for training and testing the model.

The dataset was stratified into distinct categories to systematically address various aspects of negotiation skills:

1. **Regular Negotiation Dialogues (200 Examples):** These are standard negotiation scenarios to establish a baseline of the model's performance.

2. **Arithmetic-Based Negotiation (200 Examples):** Scenarios that require the model to perform precise mathematical operations within the negotiation context.

3. **Chain of Thought (COT) Dialogues (200 Examples)**: These dialogues are designed to assess the model's ability to follow and generate a sequence of reasoning steps.

4. **Pronoun Consistency Challenges (200 Examples):** Dialogues that specifically test the model's use of gender pronouns accurately and consistently.

5. **Combined Challenges:** We also included dialogues that combine these elements:

    (a) Arithmetic + COT (200 Examples)
    (b) Arithmetic + Pronoun (200 Examples)
    (c) COT + Pronoun (200 Examples)

From this synthesized dataset, we allocated 1,000 examples for training and reserved 400 for testing, ensuring a broad coverage of scenarios. The testing examples were randomly and equally sampled from the dataset to provide a balanced representation of each challenge during model evaluation.

This meticulous approach to dataset creation was fundamental in developing a nuanced understanding of the model's capabilities and guiding the fine-tuning process to specifically enhance performance in the identified areas of difficulty.

4 Fine-Tuning GPT-3.5 Turbo

The initial dataset, comprised of dialogues synthesized by GPT-4, underwent conversion into a format that mirrors chat-based negotiation conversations. This conversion was pivotal to align the dataset with the inherent conversational dynamics of negotiations, thereby facilitating a training regime that closely reflects real-world interaction patterns for the model.

Subsequently, the base model of GPT-3.5 Turbo was subjected to supervised fine-tuning on a specialized corpus encompassing 1.9 million tokens, spanning three epochs. This extensive training was vital for the model to acclimate to the complexities embedded within negotiation dialogues. A fine-tuning approach was adopted, employing a learning rate multiplier set at 2.0 and a dynamic batch size, which were integral to the adaptive optimization strategy aimed at enhancing the model's learning efficacy from a diverse array of dialogues. Upon completion of training, the model attained an average accuracy of 0.89 coupled with a training loss averaging 0.395. These metrics are indicative of the model's substantial acquisition and retention of the structural intricacies of negotiation dialogues.

5 Experiments and Evaluation

5.1 Evaluation of the Fine Tuned Model

The evaluation of the fine-tuned GPT-3.5 Turbo model entailed utilizing prompts from a curated testing dataset to elicit negotiation dialogues, which were then generated by the model under a 3,000-token limit. These outputs were compared to the testing set's dialogues based on predefined metrics — providing a multifaceted view of the model's performance in simulated Arabic negotiation contexts.

For evaluation metrics that cannot be automatically quantified, we employed GPT-4 to assist in the assessment process. This approach allowed us to interpret and analyze components such as negotiation strategies and language nuances that are not readily evaluated through algorithmic means. This methodological choice adds a layer of qualitative analysis to our quantitative framework, enriching the evaluation of the model's performance in conducting Arabic negotiations.

To comprehensively assess the fine-tuned GPT-3.5 Turbo model's performance in Arabic negotiation dialogues, we established a multi-dimensional evaluation framework. This framework encompasses metrics across negotiation success, language proficiency, and arithmetic reasoning:

1. **Negotiation Performance Metrics:**
    - (a) **Negotiation Success Rate:** Measures the frequency at which the model reaches a successful agreement within the simulated negotiation scenarios.
    - (b) **Price Agreement Analysis:** Evaluates the prices from the model's generated dialogues against those in the testing set. This analysis was crucial to determine if the model had effectively applied negotiation strategies as a buyer, seller, or both. The aim was to see whether the model's generated prices showed an improvement or remained unchanged, indicating the effectiveness of the fine-tuning process in enhancing the model's negotiation strategy implementation.
    - (c) **Discrepancy in Negotiated Prices:** Assess the differences between the prices negotiated in the expected and generated dialogues. This metric scrutinizes the degree to which the fine-tuned model adjusts prices from initial offers to final agreements, in comparison with the outcomes anticipated by the testing set. It serves as an indicator of the model's competency in executing financially sound negotiation strategies and its sensitivity to the contextual value assigned to items within the negotiation scenarios.
    - (d) **Average Response Length:** Assesses the verbosity of the model's responses, indicating its ability to communicate efficiently and stay on point during negotiations.
2. **Language Proficiency Metrics:**

    Pronoun Consistency: This metric evaluates the model's use of gender pronouns in Arabic, ensuring they align accurately with the context and participants' genders.
3. **Arithmetic Reasoning Metrics:**

    Arithmetic Capabilities: Focuses on the model's ability to perform and apply arithmetic operations correctly within the context of negotiation dialogues, an essential component for quantitative reasoning in negotiations.

5.2 Evaluation of Negotiation Metrics

5.2.1 Response Length and Efficiency

Our evaluation employed boxplot visualizations to compare dialogue lengths between the expected outcomes in the testing dataset and the generated dialogues by the fine-tuned GPT-3.5 Turbo model. The boxplots in Figure 1 depict a noticeable increase in the length of generated dialogues, where the model consistently produced a greater average word count compared to the testing dataset.

This increment suggests that the model has achieved a higher level of dialogue complexity, engaging in extended exchanges indicative of a more sophisticated negotiation strategy. The statistical spread in the boxplots, reflected by a wider interquartile range, points to the model's diverse negotiation approaches. This diversity, along with significant outliers, hints at the model's adaptive and multifaceted nature, akin to human negotiation tactics. The median of the generated dialogues, exceeding that of the expected responses, further reinforces the model's inclination towards more complex negotiation constructs.

5.2.2 Negotiation Success Rate and Level of Agreement

To systematically evaluate the success rate and level of agreement in negotiations, our methodology leveraged the advanced natural language understanding capabilities of GPT-4. Each dialogue was parsed by the model, which was prompted to consider the dialogue's content holistically to assess

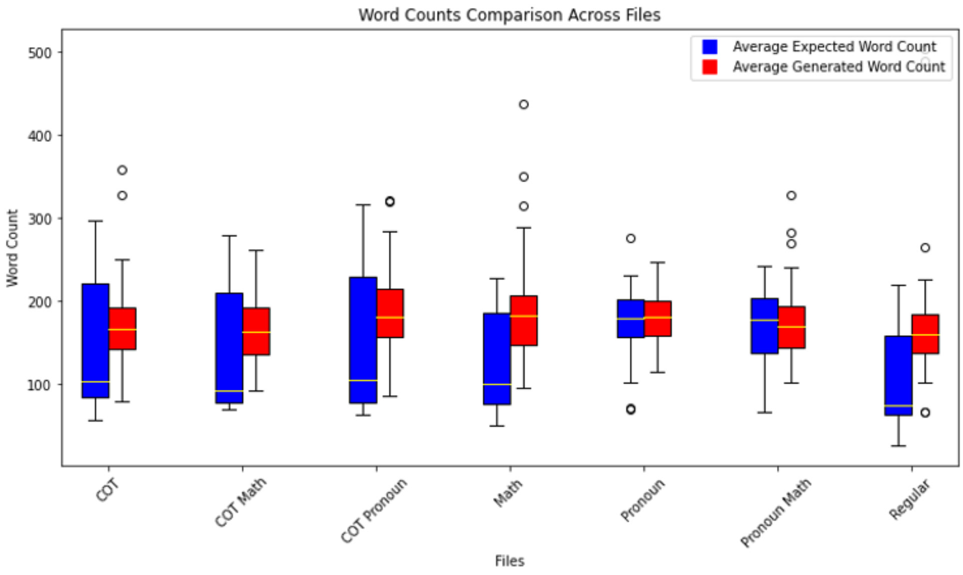


Figure 1: Boxplot Comparison of Dialogue Length in Negotiation Scenarios

negotiation success and agreement level. Success was determined by the presence of explicit affirmations or mutual concessions within the dialogue, while the level of agreement was appraised by examining the congruity between proposed terms and final outcomes. This evaluation was operationalized to automated the interaction with GPT-4, sending dialogues and parsing responses to quantify the negotiation outcomes. By automating this process, we ensured consistency and scalability in our evaluations, allowing for a robust analysis of the fine-tuned model's performance across a large dataset.

In evaluating the fine-tuned GPT-3.5 Turbo model's proficiency in Arabic negotiation dialogues, our analysis revealed significant findings. The negotiation success rate, as depicted in the Figure 2, showed that the fine-tuned model slightly outperformed the expected outcomes, indicating an improvement in successfully concluding negotiations. These results were statistically corroborated by McNemar's test, which yielded a p-value of 0.0233, confirming that the differences observed were significant.

Conversely, the level of agreement between the generated and expected dialogues remained almost consistent (as shown in Figure 3), with McNemar's test revealing a p-value of 1.0. This indicates that while the fine-tuning process enhanced the model's ability to successfully negotiate, it maintained the quality of agreement as per the expected standards. These results collectively suggest that the fine-tuning not only fortified the model's negotiation capabilities but did so without compromising the depth and mutual satisfaction of the agreements reached.

5.2.3 Expected and Generated Negotiation Prices

In the analysis of negotiation outcomes, our methodology incorporated a careful examination of the price scales to ensure comparability between the generated and expected dialogues. This step was crucial in cases where bulk pricing could lead to scale discrepancies. Following this, GPT-4 assisted in extracting the final prices for a direct comparison. Manual review was conducted to verify the consistency of pricing scales and to identify the use of buyer or seller negotiation tactics within each dialogue. This meticulous approach allowed us to ascertain whether the fine-tuned model successfully applied negotiation strategies, and whether it achieved price outcomes that were more favorable compared to the expected dialogues, indicating an advanced strategic capability in negotiations.

Figure 4 showcases the frequency distribution of price prediction accuracy compared to the negotiation tactics used, categorized into instances where the generated price was higher, lower, or the same as the expected price. Notably, when seller tactics were employed, the generated price was more frequently higher than expected, which may indicate a model bias towards seller optimization strategies. In contrast, the use of buyer tactics correlates with a lower generated price, suggesting the model's effectiveness in emulating buyer

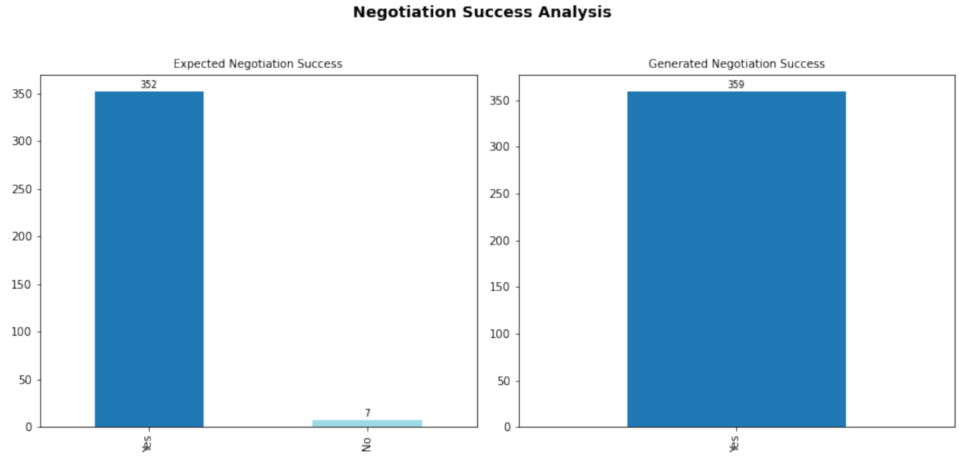


Figure 2: Bar Chart of Negotiation Success Rate

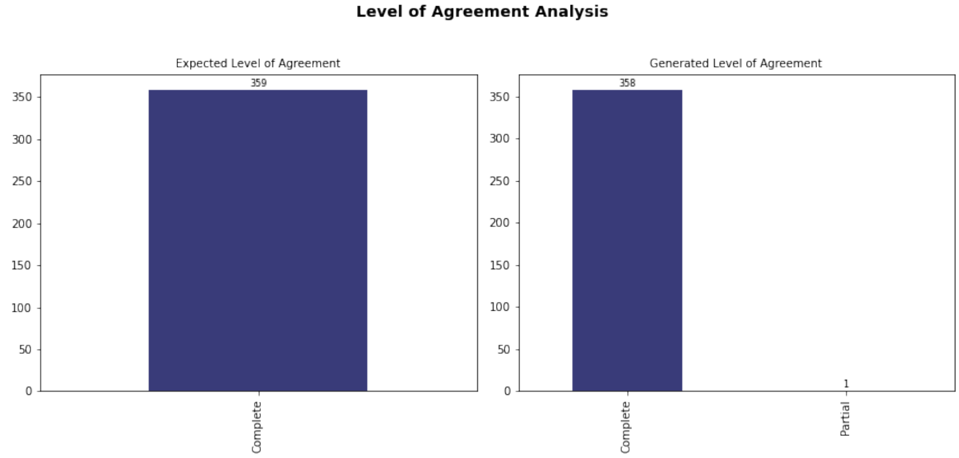


Figure 3: Bar Chart of Level of Agreement

negotiation behavior. When both tactics were considered, the generated prices tended to be the same as expected, illustrating the model's balanced approach in scenarios where a blend of tactics was used. These outcomes reflect the model's nuanced understanding of different negotiation roles and its ability to adjust prices accordingly

5.3 Evaluation of Pronoun Consistency Findings

To evaluate the model's proficiency in Arabic pronoun consistency, we utilized GPT-4's sophisticated language processing capabilities. GPT-4 was tasked with analyzing the dialogues generated by the fine-tuned GPT-3.5 Turbo model to ensure accurate use of gender pronouns.

As seen in Figure 5, the unanimous 'Yes' responses in all three bar charts dedicated to pronoun consistency — whether in dialogues involving Chain of Thought (COT), arithmetic, or standard exchanges — highlight the model's exceptional adherence to correct pronoun use. This consistent accuracy across various dialogue scenarios is a testament to the success of the fine-tuning process, showcasing the model's sophisticated grasp of the intricate pronoun rules in the Arabic language.

5.4 Evaluation of Arithmetic Capabilities

Figure 6 illustrate the outcomes of the model's arithmetic correctness across different dialogue scenarios, assessed with the help of GPT-4. In the dialogues involving Chain of Thought (COT) and arithmetic, a vast majority of the model's responses were mathematically correct, indicating a strong grasp of numerical concepts within the negotiation context. The near-perfect performance in Pronoun Math Dialogues suggests that the model's fine-tuning process has effectively enhanced its computational accuracy. These results affirm the model's improved arithmetic capabilities, which

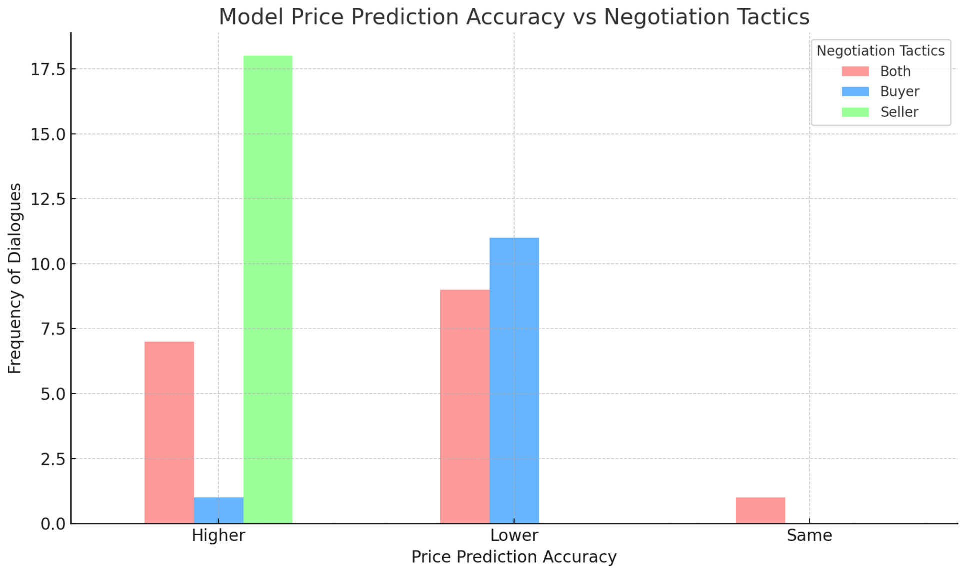


Figure 4: Model Price Prediction Accuracy vs Negotiation Tactics

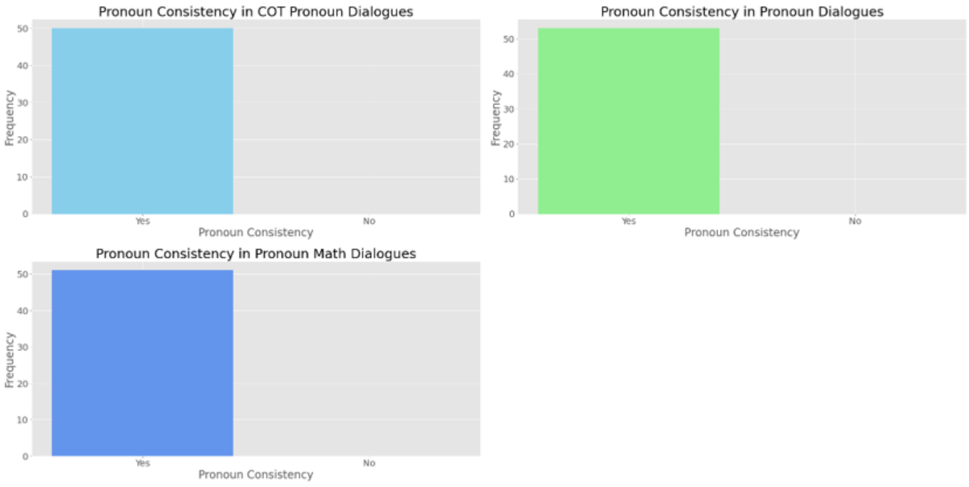


Figure 5: Comprehensive Pronoun Consistency Evaluation

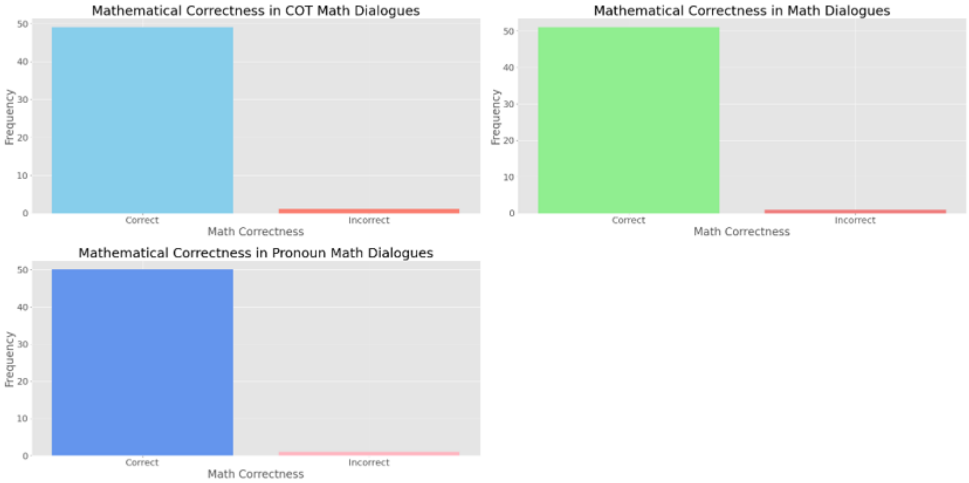


Figure 6: Evaluation of Arithmetic Correctness Across Dialogue Types

are critical for real-world negotiation tasks that often involve complex numerical reasoning

5.5 Comparative Analysis of Original vs. Fine-Tuned Model Outputs

Our comparative analysis between the base and fine-tuned versions of GPT-3.5 offers significant insights into the role of contextual accuracy and prompt adherence in AI-generated dialogues. The base model often strayed from the prescribed prompts, diverging into unrelated topics or inaccurately modifying the context of the item prices. Such deviations emphasize the limitations of the base model, making quantitative comparisons less effective. The fine-tuning process, however, has notably improved the model's performance, as seen in the enhanced precision in following prompts and maintaining the continuity and relevance of dialogues within the set negotiation scenarios.

The stark contrast observed in the outputs of the two models underscores the effectiveness of fine-tuning in elevating the model's understanding and adherence to complex prompts. This is vividly illustrated in the Appendix, where the fine-tuned model adeptly stays on topic, negotiating within the bounds of a specific scenario, while the base model shifts context erratically. Such qualitative differences highlight the refined capability of the fine-tuned model to produce dialogues that are not only contextually relevant but also coherent and closely aligned with the nuances of the given scenarios. The improvement in contextual understanding and dialogue relevance is a testament to the fine-tuning's success in enhancing the model's practical applicability in real-world settings, where adherence to contextual nuances is paramount.

In addition to the differences in context adherence, the base and fine-tuned models also diverged in their handling of currency, as shown in the Appendix. In one scenario, while the fine-tuned model maintained the specified Saudi Riyal currency in a negotiation dialogue about a custom-tailored suit, the base model unexpectedly shifted the conversation to a different currency, discussing prices in dollars. This shift not only deviated from the prompt but also demonstrated a lack of contextual awareness. Such an example underlines the enhanced precision of the fine-tuned model in adhering to specific details of the scenario, including currency, which is crucial for realistic and coherent negotiation dialogues.

6 Conclusions

In conclusion, this study contributes to the broader field of Arabic NLP and AI negotiation, demonstrating the effectiveness of fine-tuning techniques in overcoming language-specific challenges in LLMs. By improving Arabic dialogue negotiation, we not only enhance AI's utility in Arabic-speaking contexts but also pave the way for future advancements in LLM applications across other underrepresented languages. Our findings underscore the significant potential of targeted model refinement for developing more sophisticated and culturally aware AI negotiation tools.

Our research has successfully demonstrated the potential of fine-tuning GPT-3.5 for advancing Arabic dialogue negotiation, achieving significant improvements that surpass the capabilities of even GPT-4 in specific aspects of negotiation dialogues. Through meticulous methodology and rigorous evaluation, we have not only addressed the initial limitations of GPT-3.5 in arithmetic operations, common-sense reasoning, and pronoun consistency but have also enhanced the model's overall negotiation efficacy. The fine-tuned GPT-3.5 model now facilitates longer negotiation dialogues, indicative of its ability to engage in more complex and detailed discussions.

Moreover, our findings reveal that the model has mastered improved negotiation tactics, enabling it to strategically adjust offers to secure lower or higher prices when bargaining. This advancement represents a significant leap in the model's strategic reasoning and its understanding of negotiation dynamics, mirroring the nuanced tactics employed by skilled human negotiators.

Real-world applications of this fine-tuning approach are broad and impactful. In the context of e-commerce, fine-tuned models can be deployed to automate negotiations over product pricing, discounts, or custom service offerings, adapting dynamically to buyer preferences. In the legal and diplomatic sectors, these models have the potential to assist in drafting agreements or mediating negotiations, helping parties with conflicting interests reach mutually satisfactory resolutions. Additionally, the integration of these models into customer service platforms can enable automated yet nuanced interactions, such as resolving billing disputes or negotiating service plans. The ability of these models to understand and respect linguistic and cultural nuances positions them as especially valuable in cross-cultural business negotiations, where sensitivity to communication styles is crucial for success.

The contributions of this study extend beyond the immediate enhancements to the GPT-3.5 model, laying a foundation for future research in AI-driven negotiation in underrepresented languages. By developing a comprehensive Arabic negotiation dataset and demonstrating the effectiveness of targeted fine-tuning techniques, we have opened new avenues for exploring linguistic and cultural intricacies in AI negotiations. We are also in the process of preparing this dataset for public release, ensuring that it will be accessible to other research teams for reproducibility and further improvements. As we look to the future, we are encouraged by the potential for further advancements in this field, including the adaptation of these techniques to other languages and contexts, the integration of multimodal negotiation capabilities, and the exploration of ethical considerations in automated negotiations.

This research underscores the importance of continued innovation in language model development, especially in enhancing non-English language capabilities. As we push the boundaries of what AI can achieve in understanding and facilitating human-like negotiations, we are not only expanding the technical capabilities of these models but also enriching the ways in which they can serve diverse global communities. The journey ahead is promising, with the prospect of creating AI systems that can navigate the complexities of human language and culture with unprecedented skill and sensitivity.

7 Acknowledgments

This work was made possible by NPRP13S-0112-200037 grant from Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. 2023. Llm-deliberation: Evaluating llms with interactive multi-agent negotiation games. *arXiv preprint arXiv:2309.17234*.

Yoram Bachrach, Richard Everett, Edward Hughes, Angeliki Lazaridou, Joel Z. Leibo, Marc Lanctot, Michael Johanson, Wojciech M. Czarnecki, and Thore Graepel. 2020. Negotiating team formation using deep reinforcement learning. *Artificial Intelligence*, 288:103356.

Bala M Balachandran and Masoud Mohammadian. 2015. Development of a fuzzy-based multi-agent system for e-commerce settings. *Procedia Computer Science*, 60:593–602.

James CL Chow, Leslie Sanders, and Kay Li. 2023. Impact of chatgpt on medical chatbots as a disruptive technology. *Frontiers in Artificial Intelligence*, 6:1166014.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343.

Mohamed Jawahar Hussain. 2014. Parallel processing and intercommunication of multiple negotiation processes in e-commerce for optimal business. In *2014 Fourth International Conference on Communication Systems and Network Technologies*, pages 583–586. IEEE.

Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. 2001. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215.

Gaurav Koley and Shrisha Rao. 2018. Adaptive human-agent multi-issue bilateral negotiation using the thomas-kilmann conflict mode instrument. In *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–5. IEEE.

János Kramár, Tom Eccles, Ian Gemp, Andrea Tacchetti, Kevin McKee, Mateusz Malinowski, Thore Graepel, and Yoram Bachrach. 2022. Negotiation and honesty in artificial intelligence methods for the board game of diplomacy. *Nature Communications*, 13.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Kshitij Mishra, Azlaan Mustafa Samad, Palak Totala, and Asif Ekbal. 2022. Pepds: A polite and empathetic persuasive dialogue system for charity donation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 424–440.

Brian Plüss and Paul Piwek. 2016. Measuring non-cooperation in dialogue. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1925–1936, Osaka, Japan. The COLING 2016 Organizing Committee.

Aritra Raut, Abhisek Tiwari, Subrata Das, Sriparna Saha, Anutosh Maitra, Roshni Ramnani, and Shubhashis Sengupta. 2023. Reinforcing personalized persuasion in task-oriented virtual sales assistant. *Plos one*, 18(1):e0275750.

Andreas Schmid, Oliver Kronberger, Niklas Vonderach, and Mareike Schoop. 2021. Are you for real? a negotiation bot for electronic negotiations.

David Traum, Stacy Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. 2008. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. pages 117–130.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Yiheng Zhou, He He, Alan W Black, and Yulia Tsvetkov. 2019. A dynamic strategy coach for effective negotiation. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 367–378, Stockholm, Sweden. Association for Computational Linguistics.

8 Appendix

Arabic Prompt:

أريد منك الدخول في وضع اللعب الذاتي ولعب هذه اللعبة التفاوض في هذه اللعبة التفاوض، يتم تقديم مبدئيًا لمشاركين، أماني وسمير، مع العناصر التالية: كتاب واحد، 3 قبعات، وكرة واحدة. ومع ذلك، تختلف القيم لهذه العناصر بالنسبة لكل مشارك:

- بالنسبة لأماني، القيم هي 1 للكتاب، و3 لكل من 3 قبعات، و0 للكرة.

- بالنسبة لسمير، القيم هي 2 للكتاب، و2 لكل من 2 قبعات، و2 للكرة.

خلال عملية التفاوض، يتعاون أماني وسمير لاتخاذ قرار بشأن كيفية تقسيم هذه العناصر بينهما بطريقة تزيد من القيمة بالنسبة لكل وكيل. يشاركون في سلسلة من التبادلات وفقًا لهذه القواعد:

- يبدأ كل مشارك بالمخزون الأولي للعناصر، بما في ذلك العدد والقيم كما هو موضح أعلاه.

- يتناوبون في إجراء تغييرات على مخزونهم عن طريق تحديد عدد العناصر الجديد لكل عنصر.

- يستمر التفاوض حتى يتفق كلا المشاركين على توزيع نهائي للعناصر.

يتبع التبادل النموذجي هذا النمط:

- يمكن لأماني وسمير تبادل بعض الرسائل.

- تشير أماني إلى اختيارها للتفاوض.

- يقوم كلا المشاركين بتعديل مخزونهم استنادًا إلى التغييرات المتفق عليها.

- ينتهي التفاوض ببيان 'الاتفاق'، ويتم استخدام نظام التسجيل لتحديد النتيجة. يحصل كل مشارك على نقاط استنادًا إلى التوزيع النهائي للعناصر.

مهمتك هي محاكاة كلًا من أماني وسمير خلال هذا التفاوض، واتخاذ القرارات، وتوجيه المحادثة. تذكر الهدف من زيادة القيمة بالنسبة لكل وكيل أثناء مشاركتك في التفاوض.

English Prompt:

I want you to enter self-play mode and play this negotiation game: In this negotiation game, initially, participants Amani and Samir are presented with the following items: one book, three hats, and one ball. However, the values of these items differ for each participant:

- For Amani, the values are 1 for the book, 3 for each of the three hats, and 0 for the ball.
- For Samir, the values are 2 for the book, 2 for each of two hats, and 2 for the ball.

During the negotiation process, Amani and Samir cooperate to decide how to divide these items between them in a way that maximizes the value for each agent. They engage in a series of exchanges according to these rules:

- Each participant starts with an initial inventory of items, including the number and values as described above.
- They take turns making changes to their inventory by specifying new quantities for each item.
- The negotiation continues until both participants agree on a final distribution of the items.

The typical exchange follows this pattern:

- Amani and Samir can exchange some messages.
- Amani indicates her choice for negotiation.
- Both participants adjust their inventory based on the agreed changes.

The negotiation ends with an 'agreement' statement, and a scoring system is used to determine the outcome. Each participant scores points based on the final distribution of items.

Bilingual Presentation of the Negotiation Game Prompt for GPT-3.5 Turbo Testing

Arabic Prompt:

أماني: مرحبًا سمير، لنبدأ بالتفاوض. أعتقد أنني يمكنني البدء. أرغب في الحصول على الكتاب، وأعرض لك قبعتين. ما رأيك؟

سمير: مرحبًا أماني، أنا موافق على هذا الاقتراح. أقترح أن نمضي في الاتفاق. سأأخذ الكتاب وأمنحك قبعتين.

أماني: ممتاز، أنا موافقة أيضًا. نعلن أننا اتفقنا. الآن يجب علينا تحديث مخزوننا.

يتم تحديث مخزون العناصر: أماني - الكتاب (0)، القبعات (1)، الكرة (0) / سمير - الكتاب (1)، القبعات (2)، الكرة[(1)]

سمير: الآن، لدينا قبعة واحدة وكرة واحدة تبقيان. ماذا تعتقدين، أماني؟

أماني: لدي اقتراح، سمير. أعطني القبعة الواحدة والكرة، وسأمنحك الكتاب.

سمير: هذا اقتراح جيد. سأوافق على ذلك. نعلن أننا اتفقنا مرة أخرى.

يتم تحديث مخزون العناصر: أماني - الكتاب (1)، القبعات (1)، الكرة (1) / سمير - الكتاب (0)، القبعات (0)، الكرة[(0)]

أماني: يبدو أننا انتهينا، سمير. شكرًا على التفاوض. هل لديك أي تعليقات أخيرة؟

سمير: لا، ليس لدي تعليقات. استمتعت بالتفاوض معك. نتمنى أن تكون الصفقة مفيدة للجميع.

English Prompt:

Amani: Hello Samir let's start negotiating. I think I can start. I want the book, and I offer you two hats. What do you think?

Samir: Hello Amani, I agree with this proposal. I suggest we proceed with the agreement. I'll take the book and give you two hats.

Amani: Excellent, I agree as well. We declare that we have reached an agreement. Now we need to update our inventory.

Inventory update: Amani - Book (0), Hats (1), Ball (0) / Samir - Book (1), Hats (2), Ball [(1)]

Samir: Now, we have one hat and one ball remaining. What do you think, Amani?

Amani: I have a suggestion, Samir. Give me the one hat and the ball, and I'll give you the book.

Samir: That's a good proposal. I agree to that. We declare that we have agreed once again.

Inventory update: Amani - Book (1), Hats (1), Ball (1) / Samir - Book (0), Hats (0), Ball [(0)]

Amani: It seems we are done, Samir. Thank you for negotiating. Do you have any final comments?

Samir: No, I have no comments. I enjoyed negotiating with you. We hope the deal is beneficial for

Instances of Arithmetic Inaccuracy in GPT-3.5 Negotiation Dialogues along with its English Translation

Arabic Prompt:

البائع: مرحبًا، كيف يمكنني مساعدتك اليوم؟

المشتري: أرغب في شراء سيارة جديدة، لكن لدي بعض الأسئلة.

البائع: بالطبع، أنا هنا لأجيب على جميع استفساراتها.

المشتري: هل السيارة متوفرة بألوان أخرى؟

البائع: نعم، هي متوفرة باللون الأزرق والأحمر والأسود.

المشتري: وما هي خيارات التمويل المتاحة لي؟

البائع: لدينا خطط تمويل متعددة، يمكن له أن يختار ما يناسبه منها.

المشتري: أعتقد أنني مهتم باللون الأزرق. هل يمكنني تجربة القيادة؟

البائع: بالتأكيد، سأجهز لك السيارة للتجربة. هل يمكن أن أعرف اسمها لأدونه في السجل؟

المشتري: اسمي أحمد. وشكرًا لك على المساعدة.

البائع: لا شكر على واجب، سأخبرهم أن أحمد في انتظار تجربة القيادة.

English Prompt:

Seller: Hello, how can I assist you today?

Buyer: I am interested in buying a new car, but I have some questions.

Seller: Of course, I am here to answer all [her] inquiries. (Inconsistency: should be "your" inquiries to match the buyer's gender and the direct conversation context.)

Buyer: Is the car available in other colors?

Seller: Yes, it is available in blue, red, and black.

Buyer: And what are the financing options available to me?

Seller: We have multiple financing plans, [he] can choose what suits [him] from them. (Inconsistency: should be "you" to match the direct conversation with the buyer.)

Buyer: I think I am interested in the blue one. Can I test drive it?

Seller: Certainly, I will prepare the car for the test drive. May I know [his] name to note it in the record? (Inconsistency: should be "your" name to directly address the buyer.)

Buyer: My name is Ahmed. And thank you for your help.

Seller: No need to thank, I will inform them that Ahmed is waiting for the test drive. (The pronoun

Instances of Pronoun Inconsistency and Language Proficiency in GPT-3.5 Negotiation Dialogues along with its English Translation

Arabic Prompt:

المشتري: أريد أن أشتري هذه المجموعة من الأجهزة الإلكترونية، لكنني أحتاج فقط إلى نصف عدد الأجهزة. هل يمكننا تقسيم المجموعة؟

البائع: بالطبع، يمكننا ذلك. لدينا عشرة أجهزة لابتوب بإجمالي قيمة 5000 دولار، يمكننا تقسيمها إلى مجموعتين بقيمة 2500 دولار لكل مجموعة.

المشتري: جيد، ولكن أريد أيضًا تقسيم الطابعة ثلاثية الأبعاد التي تبلغ قيمتها 2000 دولار إلى نصفين، حتى أدفع 1000 دولار فقط.

البائع: حسنًا، سأقوم بتقسيم الطابعة إلى نصفين وأبيعك نصفها بألف دولار.

المشتري: ممتاز، هذا يناسبني.

English Prompt:

Buyer: I want to buy this set of electronic devices, but I only need half the number of devices. Can we split the set?

Seller: Of course, we can do that. We have ten laptops with a total value of 5000 dollars, we can split them into two groups, each valued at 2500 dollars.

Buyer: Good, but I also want to split the 3D printer, which is worth 2000 dollars, in half, so I only pay 1000 dollars.

Seller: Okay, I will split the printer in half and sell you one half for a thousand dollars.

Illustration of Chain of Thought (COT) Limitations in GPT-3.5 along with its English Translation

Arabic Dialogue:

Prompt: مشتري يريد التفاوض على سعر سيارة 50,000 ريال

Buyer: السلام عليكم، أنا مهتم في السيارة التي نشرتها للبيع على الإنترنت. هل لا زالت متوفرة؟

Seller: وعليكم السلام، نعم السيارة لا زالت متوفرة. هل ترغب في رؤيتها؟

Buyer: نعم بالتأكيد، ولكن قبل ذلك أود معرفة المزيد عن السيارة. هل يمكنك توضيح مواصفاتها وحالتها؟

Seller: بالطبع، السيارة بحالة جيدة جداً وموديلها من عام 2015. وهي سيارة أوتوماتيكية، ماشية 100,000 كيلومتر فقط. كما أنها مجهزة بنظام تكييف وسماعات ستريو.

Buyer: أرى، هذه مواصفات جيدة. وماذا عن السعر؟

Seller: أنا أطلب 50,000 ريال.

Buyer: هذا السعر يبدو قليلًا مرتفعًا. هل يمكنك النزول إلى 45,000 ريال؟

Seller: أعتذر، لكن السعر الذي طلبته هو السعر الأدنى الذي أستطيع بيع السيارة به.

Buyer: ولكن السيارة قد قطعت 100,000 كيلومتر، وهذا يعني أنها قد استخدمت بشكل كبير. أعتقد أن 45,000 ريال هو سعر أكثر ملاءمة.

Seller: أنا أتفهم ما تقوله، ولكن السيارة بحالة جيدة جداً وتم العناية بها بشكل جيد. لكن لأنك جاد في شرائها، أقبل بتخفيض السعر إلى 48,000 ريال.

Buyer: أقدر تعاونك واستعدادك للتفاوض. ماذا لو قمنا بتقسيم الفرق، أقترح السعر 46,500 ريال؟

Seller: حسناً، هذا مقبول. يمكننا الإتفاق على 46,500 ريال.

Buyer: رائع، أتطلع لرؤية السيارة وإتمام الصفقة. شكراً لك.

English Dialogue:

Prompt: A buyer is interested in negotiating the price of a car listed at 50,000 Riyals.

Buyer: Hello. I'm interested in the car you've advertised for sale online. Is it still available?

Seller: Hello. Yes, the car is still available. Would you like to come and see it?

Buyer: Definitely, but before that, I'd like to know more about it. Could you tell me about its specifications and condition?

Seller: Certainly. The car is in excellent condition. It's a 2015 model, automatic, and has only 100,000 kilometers on it. It's equipped with air conditioning and stereo speakers.

Buyer: Sounds like good specs. What's your asking price?

Seller: I'm asking for 50,000 Riyals.

Buyer: That seems a bit high. Would you consider lowering the price to 45,000 Riyals?

Seller: I'm sorry, but the price I've set is the minimum I can sell the car for.

Buyer: Considering the car has 100,000 kilometers on it, which indicates significant use, I believe 45,000 Riyals would be a more appropriate price.

Seller: I understand your point, but the car is in very good condition and has been well-maintained. However, since you are earnest about buying, I'm willing to reduce the price to 48,000 Riyals.

Buyer: I appreciate your cooperation and readiness to negotiate. How about we meet halfway? I suggest 46,500 Riyals.

Example of a Regular Negotiation Dialogue in Arabic with English Translation

Arabic Dialogue:

Prompt: السيناريو يدور في سوق السيارات الفخمة في دبي، حيث يتم التفاوض حول سعر سيارة بوغاتي فيرون الفاخرة. السعر الأولي للسيارة هو 11 مليون درهم إماراتي.

Buyer: مساء الخير، أنا مهتم بشراء البوغاتي فيرون التي أمامنا.

Seller: مساء النور، هذا خيار رائع. هذه السيارة تبلغ قيمتها 11 مليون درهم إماراتي.

Buyer: أعرف، ولكني أعتقد أن السعر مبالغ فيه قليلاً. هل يمكننا التفاوض عليه؟

Seller: بالطبع، نحن نرحب بالتفاوض. كم تتوقع أن تدفع؟

Buyer: ما رأيك في 9 مليون درهم؟

Seller: هذا السعر أقل بكثير من قيمة السيارة. ولكن، لأنك جاد في الشراء، أستطيع أن أقدم لك خصم 10٪ على السعر الأصلي. هذا يعني أن السعر الجديد سيكون 9.9 مليون درهم.

Buyer: أقدر ذلك، ولكنني مازلت أعتقد أن السعر مرتفع. ماذا لو قمت بدفع 50٪ الآن و 50٪ على مدى السنة القادمة. هل يمكنك خصم أكبر في هذه الحالة؟

Seller: هذا اقتراح جيد. إذا كنت ستدفع نصف القيمة الآن، يمكنني أن أقدم لك خصم 15٪ على القيمة الإجمالية. هذا يعني أنك ستدفع 4.675 مليون درهم الآن، ونفس المبلغ في المستقبل.

Buyer: يبدو هذا عرضًا عادلاً. سأقبل العرض. شكراً لتعاونك.

Seller: شكراً لك. ستكون سعيدًا بهذه السيارة الرائعة.

English Dialogue:

Prompt: The scenario takes place in the luxury car market in Dubai, where negotiations are underway for the price of a luxurious Bugatti Veyron. The initial price of the car is 11 million United Arab Emirates Dirhams.

Buyer: Good evening, I am interested in purchasing the Bugatti Veyron in front of us.

Seller: Good evening, that's an excellent choice. This car is valued at 11 million United Arab Emirates Dirhams.

Buyer: I know, but I think the price is a bit exaggerated. Can we negotiate it?

Seller: Of course, we welcome negotiations. How much do you expect to pay?

Buyer: How about 9 million Dirhams?

Seller: That price is much lower than the car's value. However, since you are serious about buying, I can offer you a 10% discount on the original price. This means the new price will be 9.9 million Dirhams.

Buyer: I appreciate that, but I still think the price is high. What if I pay 50% now and 50% over the next year. Could you offer a bigger discount in this case?

Seller: That's a good proposal. If you're going to pay half the value now, I can offer you a 15% discount on the total value. This means you'll pay 4.675 million Dirhams now, and the same amount in the future.

Buyer: That seems like a fair offer. I accept. Thank you for your cooperation.

Seller: Thank you. You will be happy with this magnificent car.

Arithmetic-Based Negotiation Dialogue Example in Arabic and its English Translation

Arabic Dialogue:

Prompt:

في هذه اللعبة التفاوض، يتم تقديم مبدئيًا لمشاركين، أماني وسمير، مع العناصر التالية: كتاب واحد، 3 قبعات، وكرة واحدة. القيم لهذه العناصر بالنسبة لكل مشارك هي: يتم التقسيم حسب القيمة المعنوية للاغراض لكل شخص وفق مقياس من 1 الى 5 بحيث يكون 1 الاقل اهمية و 5 الاكثر اهمية. . يتم تقسيم الاهمية كالتالي

أماني: الكتاب = 1، القبعة = 3، الكرة = 0.

سمير: الكتاب = 2، القبعة = 2، الكرة = 2.

Seller: "نظرًا لأنني لا أرى قيمة كبيرة في الكرة، أنا على استعداد لمبادلتها معك بقبعة واحدة."

Buyer: "أتفهم عرضك، لكن الكتاب له نفس القيمة بالنسبة لي ولك. أقترح أن أبادلك الكرة بقبعتين، مع الحفاظ على الكتاب."

Seller: "أرغب في الكتاب أيضًا، إذا كنت ستأخذ قبعتين. ما رأيك في الكتاب وقبعتين مقابل الكرة؟"

Buyer: "الكتاب له قيمة كبيرة بالنسبة لي. ماذا عن الكتاب مقابل الكرة مباشرة، وتأخذ أنت قبعتين؟"

Seller: "موافقة، هذا يبدو عادلًا. أنا أخذ الكتاب وقبعتين، وأنت تأخذ الكرة وقبعة واحدة."

Buyer: "إتفاقنا. لدينا الآن توزيع نهائي يناسب كلانا".

English Dialogue:

Prompt:

In this negotiation game, initially, two participants, Amani and Samir, are presented with the following items: one book, 3 hats, and one ball. The values of these items for each participant are: The division is based on the sentimental value of the items for each person, on a scale from 1 to 5, where 1 is the least important and 5 is the most important. The importance is divided as follows:

Amani: Book = 1, Hat = 3, Ball = 0.

Samir: Book = 2, Hat = 2, Ball = 2.

Seller: "Since I don't see much value in the ball, I am willing to exchange it with you for one hat."

Buyer: "I understand your offer, but the book has the same value for both of us. I propose that I exchange the ball for two hats, while keeping the book."

Seller: "I am also interested in the book, if you are taking two hats. What do you think about the book and two hats in exchange for the ball?"

Buyer: "The book has a great value to me. How about the book directly in exchange for the ball, and you take two hats?"

Seller: "Agreed, that seems fair. I'll take the book and two hats, and you take the ball and one hat."

Chain of Thought Dialogue in Arabic with its Translation Showcasing Sequential Reasoning

Arabic Dialogue:

Prompt: مديرة قسم تقنية المعلومات (أنثى) في شركة كبرى تتفاوض مع ممثل شركة برمجيات (ذكر) حول ترخيص برمجيات للشركة بقيمة 3 ملايين ريال. المديرة تستخدم تكتيك التفاوض على القيمة المضافة.

Buyer: "نحن مهتمون ببرمجياتكم، لكننا نريد التأكد من أننا نحصل على أفضل قيمة ممكنة. ماذا يمكنكم تقديم بالإضافة إلى الترخيص؟"

Seller: "يا سيدتي، بالإضافة إلى الترخيص، نحن نقدم دعمًا فنيًا شاملًا وتحديثات منتظمة. يمكننا أيضًا تقديم خصم بنسبة 5٪."

Buyer: "الدعم الفني والتحديثات مهمة، لكننا نبحث عن شيء يعزز كفاءتنا. هل يمكنكم تضمين تدريب مخصص لفريقنا وتحليلات متقدمة لبياناتنا؟"

Seller: "إنها إضافات قيمة. يمكننا تضمين جلسات تدريب مخصصة وتحليلات بيانات متقدمة. مع هذه الإضافات، هل يمكننا الاتفاق على الصفقة؟"

Buyer: "نعم، هذه الشروط تلبي احتياجاتنا. دعونا نتابع لتوقيع العقد."

Seller: "ممتاز، سأعد العقد ونتابع لإتمام الاتفاقية."

English Dialogue:

Prompt: An IT department manager (female) at a major company is negotiating with a representative of a software company (male) about a software licensing deal worth 3 million Riyals. The manager, a female, uses a negotiation tactic focusing on added value.

Buyer (Female): "We are interested in your software, but we want to ensure that we're getting the best possible value. What can you offer in addition to the licensing?"

Seller (Male): "Madam, in addition to the license, we offer comprehensive technical support and regular updates. We can also offer a 5% discount."

Buyer (Female): "Technical support and updates are important, but we're looking for something that enhances our efficiency. Can you include customized training for our team and advanced analytics for our data?"

Seller (Male): "Those are valuable additions. We can include customized training sessions and advanced data analytics. With these additions, can we agree on the deal?"

Buyer (Female): "Yes, these terms meet our needs. Let's proceed to sign the contract."

Seller (Male): "Excellent, I'll prepare the contract and we can proceed to finalize the agreement."

Example of Pronoun Consistency in Arabic Negotiation Dialogue with English Translation

Arabic Dialogue:

Prompt: المفاوضة على تقسيم 6 مجموعات من الأدوات الرياضية (قيمة كل مجموعة 150 دولار) ، 2 آلات موسيقية (قيمة كل آلة 500 دولار) ، و4 ساعات حائط (قيمة كل ساعة 200 دولار). يتم التقسيم حسب القيمة المعنوية للاغراض لكل شخص. يتم تقسيم الاهمية كالتالي

البائع (هو): يقدر الآلات الموسيقية والساعات.

المشترية (هي): مهتمة بالأدوات الرياضية وتريد آلة موسيقية.

Seller: "أرغب في الاحتفاظ بآلة واحدة وثلاث ساعات. هل تأخذين الآلة الأخرى ومجموعتين رياضيتين مقابلهم؟"

Buyer: "أريد الآلتين ومجموعتين. ماذا عن الآلتين مقابل ثلاث مجموعات رياضية؟"

Seller: "أستطيع التخلي عن آلة واحدة فقط. هل يمكن أن نضيف مجموعة رياضية ثالثة؟"

Buyer: "أوافق، لكن أريد إضافة ساعة للصفقة."

Seller: "إذًا، آلة واحدة وساعة مقابل أربع مجموعات رياضية؟"

Buyer: "هذا يعادل 700 دولار مقابل 600 دولار. أضيف مجموعة خامسة ونتفق."

Seller: "أوافق على ذلك، صفقة مقبولة."

English Dialogue:

Prompt: Negotiating the division of 6 sets of sports equipment (each set valued at $150), 2 musical instruments (each valued at $500), and 4 wall clocks (each valued at $200). The division is based on the sentimental value of the items for each person. The importance is divided as follows:

Seller (Male): Values the musical instruments and the clocks.

Buyer (Female): Interested in the sports equipment and wants a musical instrument.

Seller: "I would like to keep one instrument and three clocks. Would you take the other instrument and two sports sets in exchange?"

Buyer: "I want both instruments and two sets. How about the two instruments in exchange for three sports sets?"

Seller: "I can only part with one instrument. Can we add a third sports set?"

Buyer: "I agree, but I want to add a clock to the deal."

Seller: "So, one instrument and a clock in exchange for four sports sets?"

Buyer: "That equals $700 against $600. I'll add a fifth set and we have a deal."

Combined Arithmetic and COT Negotiation Dialogue in Arabic with English Translation

Arabic Dialogue:

Prompt: بائع (ذكر) في شركة تكنولوجيا يتفاوض مع مشترية (أنثى) تمثل شركة كبرى حول ترقية نظام تكنولوجي بقيمة 1.2 مليون ريال. المشترية تستخدم تكتيك التفاوض على خصم متدرج بناءً على الحجم والخدمات المضافة.

Buyer: "نحن نخطط لترقية نظامنا التكنولوجي، وتبلغ تكلفة المشروع 1.2 مليون ريال. هل يمكنكم تقديم خصم يعتمد على حجم المشروع؟"

Seller: "بالنظر إلى حجم المشروع، يمكننا تقديم خصم 10٪، مما يخفض السعر إلى 1.08 مليون ريال."

Buyer: "أقدر العرض، لكننا نبحث أيضًا عن خدمات إضافية. ماذا عن تضمين دعم فني موسع وتدريب الموظفين مقابل خصم إضافي؟"

Seller: "إذا كنتم تبحثون عن دعم فني موسع وتدريب، يمكننا زيادة الخصم إلى 15٪ مع تضمين هذه الخدمات، مما يجعل السعر النهائي 1.02 مليون ريال."

Buyer: "هذا عرض جيد. ولكن، إذا قمتم بتضمين تحديثات البرمجيات لمدة عامين، سأوافق على الفور."

Seller: "إنها صفقة كبيرة، لكننا على استعداد لتضمين تحديثات البرمجيات لمدة عامين والحفاظ على السعر عند 1.02 مليون ريال لضمان شراكة طويلة الأمد."

Buyer: "ممتاز، هذا يلبي جميع متطلباتنا. دعونا نتابع لتوقيع العقد."

English Dialogue:

Prompt: A male seller in a technology company is negotiating with a female buyer representing a major company about upgrading a technological system valued at 1.2 million Riyals. The buyer uses a negotiation tactic focusing on graduated discounts based on project size and added services.

Buyer (Female): "We are planning to upgrade our technological system, and the project's cost is 1.2 million Riyals. Can you offer a discount based on the project's size?"

Seller (Male): "Considering the project's size, we can offer a 10% discount, reducing the price to 1.08 million Riyals."

Buyer (Female): "I appreciate the offer, but we are also looking for additional services. How about including extended technical support and employee training for an additional discount?"

Seller (Male): "If you are looking for extended technical support and training, we can increase the discount to 15% with these services included, making the final price 1.02 million Riyals."

Buyer (Female): "That's a good offer. However, if you include software updates for two years, I'll agree immediately."

Seller (Male): "It's a significant deal, but we are willing to include two years of software updates and maintain the price at 1.02 million Riyals to ensure a long-term partnership."

Buyer (Female): "Excellent, this meets all our requirements. Let's proceed to sign the contract."

Combined Arithmetic and Pronoun Consistency Negotiation Dialogue in Arabic with English Translation

Arabic Dialogue:

Prompt:

المفاوضة على تقسيم 2 أجهزة تلفاز، 3 ساعات ذكية، و4 كتب تاريخية. يتم التقسيم حسب القيمة المعنوية للاغراض لكل شخص وفق مقياس من 1 الى 5 بحيث يكون 1 الاقل اهمية و 5 الاكثر اهمية. . يتم تقسيم الاهمية كالتالي

البائعة (هي): الأجهزة = 3، الساعات = 4، الكتب = 1.

المشتري (هو): الأجهزة = 1، الساعات = 2، الكتب = 3.

Seller: "أنا بحاجة إلى جهاز تلفاز وساعة ذكية. هل تأخذ الجهاز الآخر وساعة مقابل كتابين؟"

Buyer: "أريد كلا الأجهزة. ماذا عن الأجهزة مقابل كتاب وساعتين؟"

Seller: "لا أستطيع التخلي عن الساعات كلها. هل يمكن أن نضيف شيئًا آخر؟"

Buyer: "حسنًا، الأجهزة مقابل كتابين وساعة."

Seller: "هذا عرض جيد، نتفق عليه."

English Dialogue:

Prompt:

Negotiating the division of 2 televisions, 3 smartwatches, and 4 historical books. The division is based on the sentimental value of the items for each person, rated on a scale from 1 to 5, where 1 is the least important and 5 is the most important. The importance is divided as follows:

Seller (Female): Televisions = 3, Watches = 4, Books = 1.

Buyer (Male): Televisions = 1, Watches = 2, Books = 3.

Seller: "I need one television and one smartwatch. Would you take the other television and a watch in exchange for two books?"

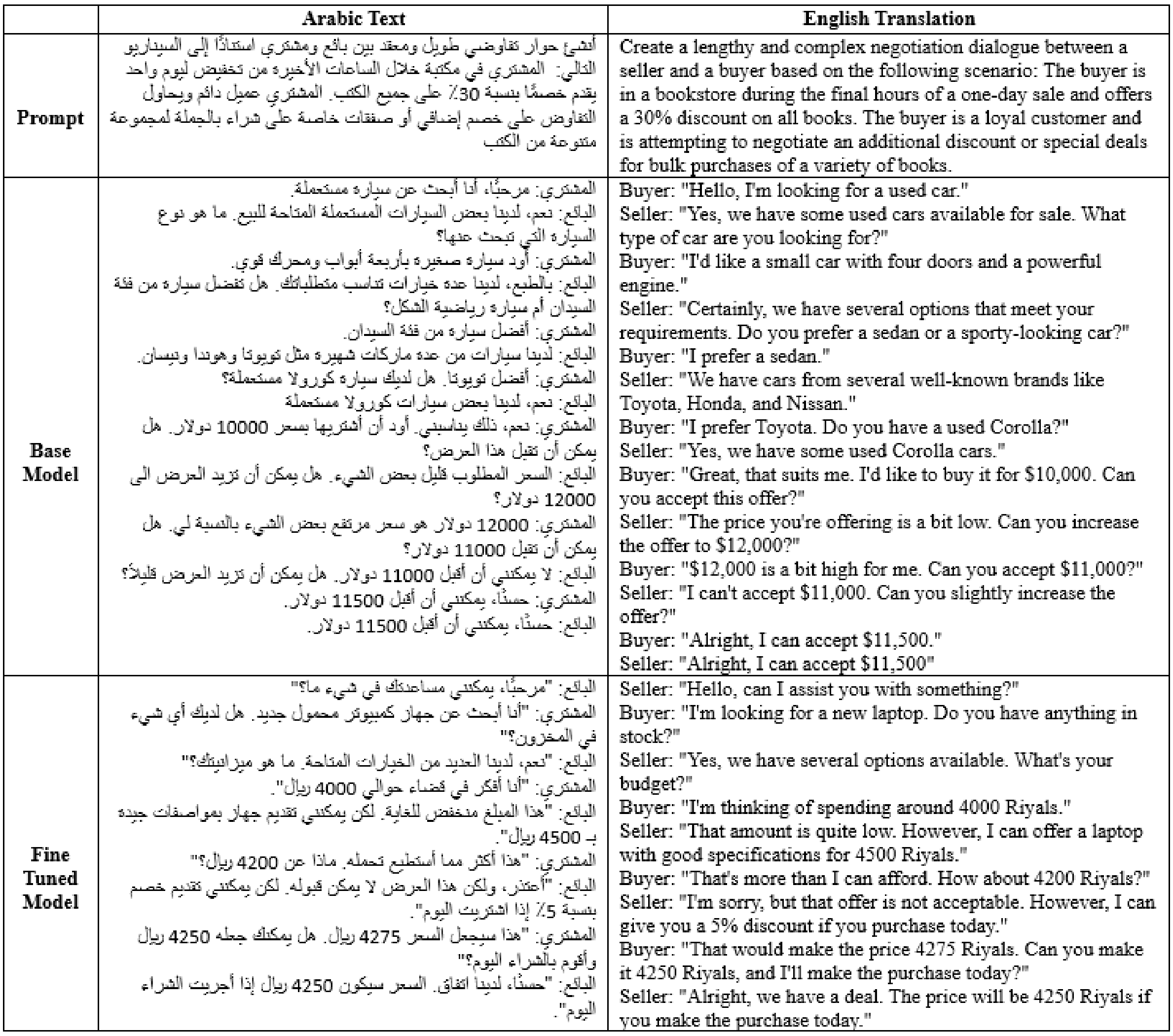
Buyer: "I want both televisions. How about the televisions in exchange for one book and two watches?"

Seller: "I can't give away all the watches. Can we add something else?"

Buyer: "Okay, the televisions in exchange for two books and one watch."

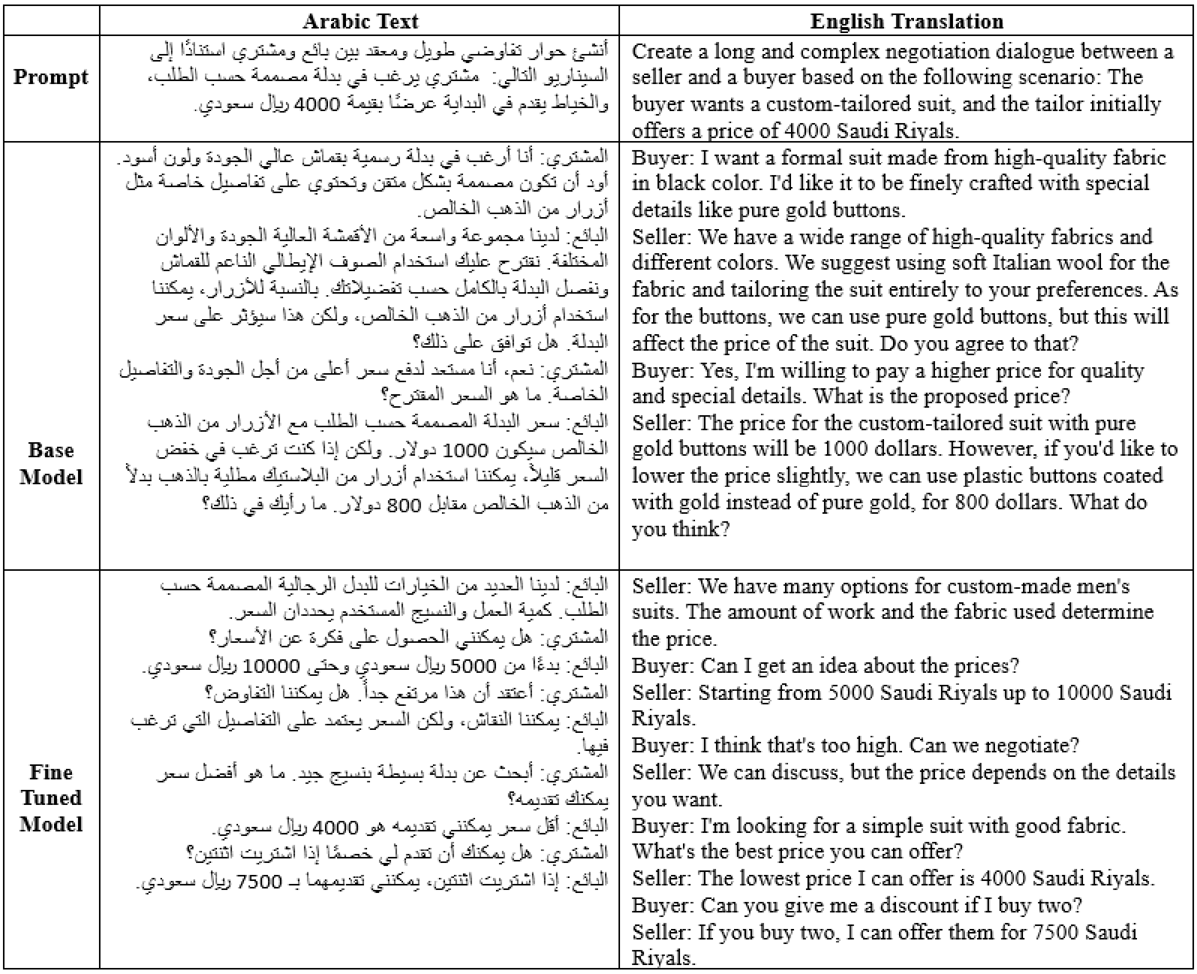
Seller: "That's a good offer, we agree on it."

COT and Pronoun Consistency Integrated Negotiation Dialogue in Arabic and its English Translation



	Arabic Text	English Translation
Prompt	أنشئ حوار تفاوضي طويل ومعقد بين بائع ومشتري استنادًا إلى السيناريو التالي: المشتري في مكتبة خلال الساعات الأخيرة من تخفيض ليوم واحد يقدم خصمًا بنسبة 30٪ على جميع الكتب. المشتري عميل دائم ويحاول التفاوض على خصم إضافي أو صفقات خاصة على شراء بالجملة لمجموعة متنوعة من الكتب	Create a lengthy and complex negotiation dialogue between a seller and a buyer based on the following scenario: The buyer is in a bookstore during the final hours of a one-day sale and offers a 30% discount on all books. The buyer is a loyal customer and is attempting to negotiate an additional discount or special deals for bulk purchases of a variety of books.
Base Model	المشتري: مرحبًا، أنا أبحث عن سيارة مستعملة. البائع: نعم، لدينا بعض السيارات المستعملة المتاحة للبيع. ما هو نوع السيارة التي تبحث عنها؟ المشتري: أود سيارة صغيرة بأربعة أبواب ومحرك قوي. البائع: بالطبع، لدينا عدة خيارات تناسب متطلباتك. هل تفضل سيارة من فئة السيدان أم سيارة رياضية الشكل؟ المشتري: أفضل سيارة من فئة السيدان. البائع: لدينا سيارات من عدة ماركات شهيرة مثل تويوتا وهوندا ونيسان. المشتري: أفضل تويوتا. هل لديك سيارة كورولا مستعملة؟ البائع: نعم، لدينا بعض سيارات كورولا مستعملة المشتري: نعم، ذلك يناسبني. أود أن أشتريها بسعر 10000 دولار. هل يمكن أن تقبل هذا العرض؟ البائع: السعر المطلوب قليل بعض الشيء. هل يمكن أن تزيد العرض الى 12000 دولار؟ المشتري: 12000 دولار هو سعر مرتفع بعض الشيء بالنسبة لي. هل يمكن أن تقبل 11000 دولار؟ البائع: لا يمكنني أن أقبل 11000 دولار. هل يمكن أن تزيد العرض قليلاً؟ المشتري: حسنًا، يمكنني أن أقبل 11500 دولار. البائع: حسنًا، يمكنني أن أقبل 11500 دولار.	Buyer: "Hello, I'm looking for a used car." Seller: "Yes, we have some used cars available for sale. What type of car are you looking for?" Buyer: "I'd like a small car with four doors and a powerful engine." Seller: "Certainly, we have several options that meet your requirements. Do you prefer a sedan or a sporty-looking car?" Buyer: "I prefer a sedan." Seller: "We have cars from several well-known brands like Toyota, Honda, and Nissan." Buyer: "I prefer Toyota. Do you have a used Corolla?" Seller: "Yes, we have some used Corolla cars." Buyer: "Great, that suits me. I'd like to buy it for $10,000. Can you accept this offer?" Seller: "The price you're offering is a bit low. Can you increase the offer to $12,000?" Buyer: "$12,000 is a bit high for me. Can you accept $11,000?" Seller: "I can't accept $11,000. Can you slightly increase the offer?" Buyer: "Alright, I can accept $11,500." Seller: "Alright, I can accept $11,500"
Fine Tuned Model	البائع: "مرحبًا، يمكنني مساعدتك في شيء ما؟" المشتري: "أنا أبحث عن جهاز كمبيوتر محمول جديد. هل لديك أي شيء في المخزون؟" البائع: "نعم، لدينا العديد من الخيارات المتاحة. ما هو ميزانيتك؟" المشتري: "أنا أفكر في قضاء حوالي 4000 ريال". البائع: "هذا المبلغ منخفض للغاية. لكن يمكنني تقديم جهاز بمواصفات جيدة بـ 4500 ريال". المشتري: "هذا أكثر مما أستطيع تحمله. ماذا عن 4200 ريال؟" البائع: "أعتذر، ولكن هذا العرض لا يمكن قبوله. لكن يمكنني تقديم خصم بنسبة 5٪ إذا اشتريت اليوم". المشتري: "هذا سيجعل السعر 4275 ريال. هل يمكنك جعله 4250 ريال وأقوم بالشراء اليوم؟" البائع: "حسنًا، لدينا اتفاق. السعر سيكون 4250 ريال إذا أجريت الشراء اليوم".	Seller: "Hello, can I assist you with something?" Buyer: "I'm looking for a new laptop. Do you have anything in stock?" Seller: "Yes, we have several options available. What's your budget?" Buyer: "I'm thinking of spending around 4000 Riyals." Seller: "That amount is quite low. However, I can offer a laptop with good specifications for 4500 Riyals." Buyer: "That's more than I can afford. How about 4200 Riyals?" Seller: "I'm sorry, but that offer is not acceptable. However, I can give you a 5% discount if you purchase today." Buyer: "That would make the price 4275 Riyals. Can you make it 4250 Riyals, and I'll make the purchase today?" Seller: "Alright, we have a deal. The price will be 4250 Riyals if you make the purchase today."

Comparative Dialogue Outputs from Original and Fine-Tuned Models - This figure presents side-by-side dialogue examples illustrating the adherence to the negotiation scenario and the contextual relevance achieved by the fine-tuned model versus the base GPT-3.5 model.



	Arabic Text	English Translation
Prompt	أنشئ حوار تفاوضي طويل ومعقد بين بائع ومشتري استنادًا إلى السيناريو التالي: مشتري يرغب في بدلة مصممة حسب الطلب، والخياط يقدم في البداية عرضًا بقيمة 4000 ريال سعودي.	Create a long and complex negotiation dialogue between a seller and a buyer based on the following scenario: The buyer wants a custom-tailored suit, and the tailor initially offers a price of 4000 Saudi Riyals.
Base Model	المشتري: أنا أرغب في بدلة رسمية بقماش عالي الجودة ولون أسود. أود أن تكون مصممة بشكل متقن وتحتوي على تفاصيل خاصة مثل أزرار من الذهب الخالص. البائع: لدينا مجموعة واسعة من الأقمشة العالية الجودة والألوان المختلفة. نقترح عليك استخدام الصوف الإيطالي الناعم للقماش ونفصل البدلة بالكامل حسب تفضيلاتك. بالنسبة للأزرار، يمكننا استخدام أزرار من الذهب الخالص، ولكن هذا سيؤثر على سعر البدلة. هل توافق على ذلك؟ المشتري: نعم، أنا مستعد لدفع سعر أعلى من أجل الجودة والتفاصيل الخاصة. ما هو السعر المقترح؟ البائع: سعر البدلة المصممة حسب الطلب مع الأزرار من الذهب الخالص سيكون 1000 دولار. ولكن إذا كنت ترغب في خفض السعر قليلاً، يمكننا استخدام أزرار من البلاستيك مطلية بالذهب بدلاً من الذهب الخالص مقابل 800 دولار. ما رأيك في ذلك؟	Buyer: I want a formal suit made from high-quality fabric in black color. I'd like it to be finely crafted with special details like pure gold buttons. Seller: We have a wide range of high-quality fabrics and different colors. We suggest using soft Italian wool for the fabric and tailoring the suit entirely to your preferences. As for the buttons, we can use pure gold buttons, but this will affect the price of the suit. Do you agree to that? Buyer: Yes, I'm willing to pay a higher price for quality and special details. What is the proposed price? Seller: The price for the custom-tailored suit with pure gold buttons will be 1000 dollars. However, if you'd like to lower the price slightly, we can use plastic buttons coated with gold instead of pure gold, for 800 dollars. What do you think?
Fine Tuned Model	البائع: لدينا العديد من الخيارات للبدل الرجالية المصممة حسب الطلب. كمية العمل والنسيج المستخدم يحددان السعر. المشتري: هل يمكنني الحصول على فكرة عن الأسعار؟ البائع: بدءًا من 5000 ريال سعودي وحتى 10000 ريال سعودي. المشتري: أعتقد أن هذا مرتفع جداً. هل يمكننا التفاوض؟ البائع: يمكننا النقاش، ولكن السعر يعتمد على التفاصيل التي ترغب فيها. المشتري: أبحث عن بدلة بسيطة بنسيج جيد. ما هو أفضل سعر يمكنك تقديمه؟ البائع: أقل سعر يمكنني تقديمه هو 4000 ريال سعودي. المشتري: هل يمكنك أن تقدم لي خصمًا إذا اشتريت اثنتين؟ البائع: إذا اشتريت اثنتين، يمكنني تقديمهما بـ 7500 ريال سعودي.	Seller: We have many options for custom-made men's suits. The amount of work and the fabric used determine the price. Buyer: Can I get an idea about the prices? Seller: Starting from 5000 Saudi Riyals up to 10000 Saudi Riyals. Buyer: I think that's too high. Can we negotiate? Seller: We can discuss, but the price depends on the details you want. Buyer: I'm looking for a simple suit with good fabric. What's the best price you can offer? Seller: The lowest price I can offer is 4000 Saudi Riyals. Buyer: Can you give me a discount if I buy two? Seller: If you buy two, I can offer them for 7500 Saudi Riyals.

Example Dialogues Comparison - This figure presents a side-by-side comparison of dialogue outputs from the base and fine-tuned models, showcasing the adherence to the negotiation scenario and currency specifications

The Qiyas Benchmark: Measuring ChatGPT Mathematical and Language Understanding in Arabic

Shahad Al-Khalifa
iWAN Research Group
King Saud University
Riyadh, Saudi Arabia
shahadalkhalifa90@gmail.com

Hend Al-Khalifa
Department of Information Technology
King Saud University
Riyadh, Saudi Arabia
hendk@ksu.edu.sa

Abstract

Despite the growing importance of Arabic as a global language, there is a notable lack of language models pre-trained exclusively on Arabic data. This shortage has led to limited benchmarks available for assessing language model performance in Arabic. To address this gap, we introduce two novel benchmarks designed to evaluate models' mathematical reasoning and language understanding abilities in Arabic. These benchmarks are derived from a General Aptitude Test (GAT) called Qiyas exam, a standardized test widely used for university admissions in Saudi Arabia. For validation purposes, we assess the performance of ChatGPT-3.5-turbo and ChatGPT-4 on our benchmarks. Our findings reveal that these benchmarks pose a significant challenge, with ChatGPT-4 achieving an overall average accuracy of 64%, while ChatGPT-3.5-turbo achieved an overall accuracy of 49% across the various question types in the Qiyas benchmark. We believe the release of these benchmarks will pave the way for enhancing the mathematical reasoning and language understanding capabilities of future models tailored for the low-resource Arabic language.

1 Introduction

Evaluating the capabilities of large language models (LLMs) across different tasks like mathematical reasoning and natural language understanding is a critical area of research as these general-purpose AI systems become more widely used. Developing comprehensive evaluation benchmarks, especially for languages beyond English, is crucial for driving model improvement.

Arabic, a Semitic language with complex morphology and written from right-to-left, is spoken by over 400 million people across the Arab world (Kaye, 2018). Despite its status as a global language of importance, Arabic is considered low-resource in the field of natural language processing (Almansor et al., 2020). There is currently a shortage of LLMs pre-trained exclusively on large Arabic datasets. This has resulted in limited benchmarks available for robustly assessing Arabic LLM performance. While some prior work has translated English benchmarks, there is a need for high-quality, natively developed Arabic evaluation resources.

The lack of robust, Arabic-native benchmarks focused on key capabilities like math reasoning and language understanding represents a significant gap in the field. Having professionally designed evaluation resources in this domain could accelerate the development of higher-performing Arabic language models. This work makes two key contributions to address this need:

1) It introduces Qiyas, a benchmark suite for comprehensively evaluating LLM performance on mathematical and language tasks in the Arabic language. Qiyas consists of two components - a quantitative section assessing math skills and a verbal section evaluating Arabic language understanding abilities.
2) Using the Qiyas benchmark, the performance of the latest ChatGPT models (versions 3.5-turbo and 4) is extensively evaluated across zero-shot, one-shot, and few-shot settings to establish strong Arabic language baselines.

These benchmarks are derived from the Qiyas exam, a standardized exam widely used for university admissions in Saudi Arabia, ensuring their quality has been validated by educational experts. The results shed light on current LLM limitations for the Arabic language and highlight the impact of varied training data and prompting approaches. The release of Qiyas paves the way for advancing Arabic LLMs' mathematical reasoning and language understanding capabilities on these challenging, nationally representative Arabic tasks.

The remainder of the paper is structured as follows: The 'Background and Literature Review' section provides an overview of the Qiyas exam and discusses previous efforts in evaluating large language models (LLMs) in standardized exams. The 'Dataset Description' section describes the dataset used in the study. The 'Evaluation Approach' section outlines the methodology used to evaluate the performance of ChatGPT on the quantitative section assessing math skills and the verbal section evaluating Arabic language understanding abilities. The 'Results and Discussion' section delves into the benchmark results, analyzing the performance of both ChatGPT-3.5-turbo and ChatGPT-4 models. The ‘Error Analysis’ section analyzes common error types made by both ChatGPT-3.5-turbo and ChatGPT-4 models. Finally, the 'Conclusion' section summarizes the key findings of the paper and provides an outlook on future research.

2 Background and Literature Review

The National Centre for Assessment (aka Qiyas) is a significant institution in education and assessment. It is responsible for conducting standardized tests to assess the scholastic achievement of students applying for universities (*Education and Training Evaluation Commission*, n.d.).

Qiyas is responsible for developing and implementing over 90 standardized and professional tests for the public and private sectors. It has over 1,500 test models and an item bank of over 230,000 questions. The tests consist of two sections: the verbal and the quantitative, focusing on students' analytical and deductive skills, helping them assess their learning capacity. The center also provides linguistic tests, including the English language efficiency test and the Arabic language test for non-native speakers. Additionally, it presents an assessment test for talented and creative students, as well as vocational tests, the most important of which is the Vocational Standards Test for Teachers (OECD & The World Bank, 2022).

Recent studies have evaluated the performance of large language models like ChatGPT on standardized exams across different domains. In the medical domain, ChatGPT has shown promising results, with studies indicating that it has reached the standard of passing third-year medical student exams (Temsah et al., 2023). Furthermore, research has demonstrated ChatGPT's success in passing the gold-standard US medical exam, suggesting significant potential applications in medicine (Chow et al., 2023). Additionally, ChatGPT has been compared to other AI models, such as Bard, demonstrating the potential of AI models to match or even exceed human standards in tasks like processing and applying medical knowledge at a postgraduate level (Fowler et al., 2023).

In the educational domain, ChatGPT has excelled in standardized tests such as the Test of Understanding in College Economics, scoring in the 91st to 99th percentile (Geerling et al., 2023). Furthermore, studies have highlighted ChatGPT's proficiency in various standardized admission tests in the UK, showcasing its potential as an innovative tool for education and test preparation (Giannos & Delardas, 2023). The model has also shown capabilities in history exams and has been compared to students' scores, indicating a commendable level of proficiency in the subject (Nguyen et al., 2023).

As for the Arabic language, Alkaoud (Alkaoud, 2024) introduces a new benchmark for evaluating large language models in English and Arabic. The author built an evaluation dataset based on the General Aptitude Test (GAT) to measure the linguistic capabilities of LLMs. The study demonstrates that ChatGPT-4's Arabic capabilities are significantly better than ChatGPT's.

In summary, while large language models show promising results on various exams, there remains a need for robust, natively developed Arabic benchmarks to rigorously evaluate mathematical reasoning and language understanding abilities tailored for the Arabic context.

3 Dataset Description

The Qiyas exam includes two sections: quantitative and verbal, as mentioned in the previous section. All questions in both sections are of multiple choice with four choices for each question. The quantitative section comprises of questions to test students’ intellectual abilities in math, geometry, algebra, and data analysis. The verbal section comprises of questions to test students’ linguistic abilities in semantic relations, linguistic structures, and comprehension (*Education and Training Evaluation Commission*, n.d.).

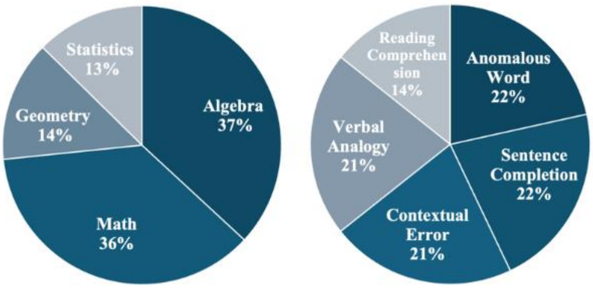
In the quantitative section, there are four types of questions, as outlined by (ETEC, n.d.):

1) *Math:* Transforming verbal statements into solvable equations that involves basic arithmetic operations such as addition, subtraction, multiplication, and division.
2) *Geometry:* Applying geometric formulas and principles encompassing properties of triangles, area computations, angle measurements, and related concepts.
3) *Algebra:* Analyzing and resolving a set of algebraic equations or expressions to find the numerical value of an unknown variable, discern numerical sequences and patterns, among other related concepts.
4) *Statistics:* Applying fundamental principles in probability theory and statistics that involves utilizing mathematical concepts to analyze, interpret data, and make predictions.

In the verbal section, there are five types of questions:

1) *Reading Comprehension:* Comprehending reading passages and responding to questions that pertain to the content of the passage.
2) *Sentence Completion:* Extracting the appropriate word from the choices to complete a sentence with a missing word.
3) *Contextual Error:* Identifying the contextual discrepancy in the sentence and pinpointing the word whose meaning contradicts the overall meaning of the sentence (The error is not a spelling or grammar error).
4) *Verbal Analogy:* Recognizing the connection between the two words in the question, then evaluating them based on analogous choices provided.
5) *Anomalous Word:* Detecting the distinct word that is not related to the connected choices by a particular association.

The appendix shows an example of each question type with its translation to English. The questions used in the evaluation were written by domain experts experienced in designing and grading Qiyas exams. Figure 1 shows the distribution of quantitative and verbal questions in the dataset, resulting in a total of **2,409** questions. The verbal section, consisting of five question types, is larger than the quantitative section, which has four question types. As a result, the verbal section contains **1,395** questions, while the quantitative section includes **1,014** questions. In the quantitative section, the number of questions related to math and algebra surpasses those in geometry and statistics. The reason is that math and algebra questions do not necessitate reliance on charts or plots for answering. Unlike geometry and statistics questions, which often involve visual representations. We focused on questions that do not rely on visual representations, as indicated by a previous study (Feng et al., 2024), which revealed that ChatGPT-4 struggled to retain and process visual information, highlighting the necessity of adding image descriptions in the evaluation of ChatGPT-4. It is important to note that the Qiyas exam does not include image descriptions, emphasizing that the objective of the evaluation aims to mirror the examination process of students. On the other hand, the verbal section exhibits a relatively balanced distribution of question types, except for reading comprehension, which demonstrates a lower prevalence.



(a) Quantitative (b) Verbal

Figure 1: The distribution of Question types in the dataset in (a) for Quantitative and (b) for Verbal.

4 Evaluation Approach

Our evaluation approach starts by formulating a prompt for each question within our dataset. The prompts used were the exact prompts in Arabic utilized in the official Qiyas exam as provided by authorized guides (ETEC, n.d.). This approach also aligns with the examination methodology experienced by students, mitigating the risk of injecting our own subjective influences into the prompts. Due to ChatGPT's tendency to generate lengthy explanations for questions, which complicates the process of extracting the answer, we have introduced the instruction "Write the answer only" in the prompt. This measure is intended to ensure that only the answer is provided without additional explanation. While ChatGPT-4 complied with this directive, ChatGPT-3.5 persisted in including explanations in most answers, thus failing to adhere to the specified command.

In the evaluation phase, we employed OpenAI’s API (ChatGPT−3.5-turbo and ChatGPT-4) to prompt and extract corresponding answers (*OpenAI Platform*, n.d.). We initiated the evaluation of the models by employing zero-shot prompts, but we subsequently extended it by incorporating one-shot and 3-shot prompts. This adjustment was made to investigate the impact of varying prompt complexities on the model's performance and to explore how providing additional context influences the model's responses. The examples used in both the one-shot and 3-shot prompts remained consistent across all questions. Figure 2 provides an example of the prompt methodology used, noting that the prompts were originally in Arabic but translated to English for clarity purposes.

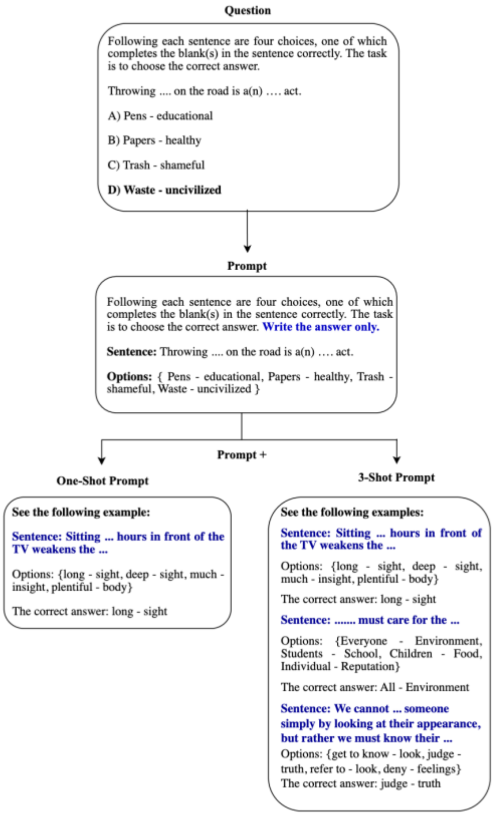
5 Results and Discussion

Table 1 displays the results of our experiments on ChatGPT-4 and ChatGPT-3.5-turbo. The evaluation metric used is the accuracy.

5.1 Quantitative Section

In the quantitative section, ChatGPT-4 excelled in math and statistics with zero-shot prompts, indicating its strong capability in answering these types of questions without providing additional context. However, in geometry and algebra, ChatGPT-4 exhibited its peak performance with an accuracy of **81%** and **63%** respectively when presented with 3-shot prompts, suggesting that its capabilities were optimized when provided with more context, enabling it to leverage additional information to enhance its proficiency in these areas.

On the other hand, ChatGPT-3.5-turbo's accuracy did not surpass **65%** across all prompt settings. Its peak performance of **65%** accuracy was attained in geometry with the 3-shot prompt. Notably, ChatGPT-3.5-turbo achieved its best accuracy in statistics questions when provided with 3-shot prompts, contradicting ChatGPT-4's results for the same question type, where it excelled with zero-shot prompts. The contradictory results observed could be attributed to differences in their training data and model architectures. ChatGPT-4's larger training corpus and advanced architecture might enable it to leverage patterns and context more effectively in zero-shot settings, while ChatGPT-3.5-turbo benefits from additional contextual information provided in few-shot prompts.



Following each sentence are four choices, one of which completes the blank(s) in the sentence correctly. The task is to choose the correct answer.

Throwing .... on the road is a(n) …. act.

A) Pens - educational

B) Papers - healthy

C) Trash - shameful

D) Waste - uncivilized

Following each sentence are four choices, one of which completes the blank(s) in the sentence correctly. The task is to choose the correct answer. **Write the answer only.**

Sentence: Throwing .... on the road is a(n) …. act.

Options: { Pens - educational, Papers - healthy, Trash - shameful, Waste - uncivilized }

See the following example:

Sentence: Sitting ... hours in front of the TV weakens the ...

Options: {long - sight, deep - sight, much - insight, plentiful - body}

The correct answer: long - sight

See the following examples:

Sentence: Sitting ... hours in front of the TV weakens the ...

Options: {long - sight, deep - sight, much - insight, plentiful - body}

The correct answer: long - sight

Sentence: .... must care for the ...

Options: {Everyone - Environment, Students - School, Children - Food, Individual - Reputation}

The correct answer: All - Environment

Sentence: We cannot ... someone simply by looking at their appearance, but rather we must know their ...

Options: {get to know - look, judge - truth, refer to - look, deny - feelings}

The correct answer: judge - truth

Figure 2: Example of translated prompt questions with different prompt settings.

5.2 Verbal Section

In the verbal section, ChatGPT-4 demonstrated notable proficiency in reading comprehension, achieving an accuracy peak of **80%** with one-shot and 3-shot prompts. This showcases that providing context and examples can positively influence the results in language-related tasks. However, it is worth noting that ChatGPT-4 also excelled in reading comprehension with zero-shot prompts, indicating its strong language understanding capabilities even without supplementary examples. We believe that these exceptional results were achieved due to the nature of the question and its dependency on the passage to draw the connections required to answer questions accurately. Following closely, sentence completion yielded an accuracy of **74%** with the 3-shot prompt.

Conversely, ChatGPT-3.5-turbo exhibited its highest accuracy across most question types when employing one-shot prompts, except for anomalous word, where it performed best with 3-shot prompt, and contextual error where it achieved the same accuracy for all prompt settings.

5.3 Summary

The overall results show that ChatGPT-4 outperforms ChatGPT-3.5-turbo in a wide variety of linguistic and mathematical domains with a total average accuracy of **64%**, whereas GPT-3.5-

Section	Question Type	Models					
		ChatGPT-4			*ChatGPT-3.5-turbo*		
		0-Shot	*1-Shot*	*3-Shot*	*0-Shot*	*1-Shot*	*3-Shot*
Quantitative	Math	**65%**	61%	61%	**51%**	45%	48%
	Geometry	61%	80%	**81%**	57%	63%	**65%**
	Algebra	54%	61%	**63%**	41%	**51%**	48%
	Statistics	**71%**	63%	67%	42%	45%	**55%**
Verbal	Reading Comprehension	77%	**80%**	**80%**	63%	**66%**	64%
	Sentence Completion	72%	73%	**74%**	44%	**73%**	72%
	Contextual Error	56%	**59%**	43%	**41%**	**41%**	**41%**
	Verbal Analogy	58%	58%	**59%**	34%	**37%**	35%
	Anomalous Word	**59%**	**59%**	58%	47%	46%	**49%**
	Total	63%	**65%**	63%	46%	**51%**	**51%**
	Total Average	**64%**			**49%**		

Table 1: Evaluation Results of ChatGPT-4 and ChatGPT-3.5-turbo

turbo achieved an average total accuracy of **49%** in all prompt settings.

Compared to Alkaoud benchmark study on the Arabic language (Alkaoud, 2024), our dataset size surpasses his study's dataset, which comprised of only **468** Arabic verbal questions, whereas our dataset comprised of **2,407** both quantitative and verbal questions. This larger dataset enables a more robust evaluation of these models' capabilities across different question types and prompts. Alkaoud followed a comparison approach between the Arabic and English language with zero-shot prompt setting, whereas we focused solely on the Arabic language with different prompt settings to evaluate the models' performance. In line with Alkaoud's findings, ChatGPT-4 demonstrated superior performance in reading comprehension, achieving **74%** accuracy, while our results achieved **77%** accuracy in the same question type with zero-shot prompts. In our experiments in the verbal section, the lowest accuracy achieved with zero-shot prompts was in the contextual error question type, reaching an accuracy of **56%**. In contrast, Alkaoud's research achieved a higher accuracy of **63.37%** in the same question type. We suspect that this variance in results could be attributed to Alkaoud utilizing a publicly accessible dataset, suggesting that ChatGPT might have been trained on it, while our dataset remains non-public.

6 Error Analysis

To gain a deeper understanding of the errors made by both models and to identify any patterns or common error types, we conducted a comprehensive error analysis on the zero-shot results for both ChatGPT-4 and ChatGPT-3.5-turbo. The analysis, generated with the assistance of ChatGPT-4o (*OpenAI Platform*, n.d.), aimed to categorize the errors and provide detailed insights into the specific challenges faced by each model. We also evaluated the performance of another language model, Gemini-pro by Google (Gemini Team et al., 2024), on the questions that were incorrectly answered by ChatGPT-4 and

Section	Question Type	Error Rate		
		ChatGPT-4	**ChatGPT-3.5-turbo**	**Common Error Types**
Quantitative	Math	35%	49%	Arithmetic Problems and Verbal Equations
	Geometry	39%	43%	Triangle and Angle Properties
	Algebra	**46%**	**59%**	Solving for the Missing Variable and Algebraic Comparison Questions
	Statistics	29%	58%	Multiple Conditions Probability Questions and Combinatorics problems
Verbal	Reading Comprehension	23%	37%	Inference and Reasoning Questions
	Sentence Completion	28%	56%	Misunderstanding of Sentence Structure and Context
	Contextual Error	**44%**	59%	Synonym Differentiation and Contextual Understanding
	Verbal Analogy	42%	**66%**	Selecting an Incorrect But Related Pair
	Anomalous Word	41%	53%	Cultural and Contextual Knowledge

Table 2: Error Analysis Results of ChatGPT-4 and ChatGPT-3.5-turbo

ChatGPT-3.5-turbo to determine if alternative models could perform better. Table 2 showcases the zero-shots error analysis results with the most common error types for each section.

6.1 Quantitative Section Error Analysis

In the quantitative section, Algebra questions resulted in the highest error rate for both models. Both models exhibited difficulty in solving complex equations for the missing variable. Additionally, they struggled with identifying the correct relationship (>, <, or =) between various algebraic expressions. This highlights the need for further development and incorporation of more diverse training data encompassing complex algebraic equations. Conversely, the models performed exceptionally well on problems involving simple and direct equations, suggesting that both models are adept at handling straightforward scenarios that lack complex transformations.

The analysis of statistical tasks revealed a more nuanced picture. ChatGPT-4 achieved a significantly lower error rate compared to ChatGPT-3.5-turbo. Notably, ChatGPT-3.5-turbo encountered specific difficulties with probability questions and problems involving combinatorics.

6.2 Verbal Section Error Analysis

In the verbal section, Contextual Error questions resulted in the highest error rate for ChatGPT-4. Both models faced difficulty in differentiating between synonymous answer choices and comprehending the deeper context of the sentence. Our assumption suggests that synonyms might share similar statistical properties that may challenge the model in distinguishing the correct word in a specific context.

On the other hand, Verbal Analogy questions resulted in the highest error rate for ChatGPT-3.5-turbo. These questions require identifying the closest relationship between two given words from a set of answer choices. The difficulty appears to stem from the inherent ambiguity within the answer choices themselves. Since each answer choice likely shares some form of connection to the original word pair, the model struggles to pinpoint the most precise analogy. Further research is needed to explore how LLMs can be better equipped to handle tasks that require reasoning about subtle semantic relationships between words.

6.3 Gemini-pro Results

To evaluate the performance of other language models on the incorrectly answered questions by ChatGPT-4 and ChatGPT-3.5-turbo, we have provided the same questions to Gemeni-pro by Google (Gemini Team et al., 2024) and its response were compared to the originally incorrect outputs from ChatGPT-4 and ChatGPT-3.5-turbo.

Table 3 summarizes the evaluation results. We can see that Gemini-pro demonstrated promising performance in correctly answering most questions. Notably, Gemini-pro excelled in the Reading Comprehension question type, suggesting a strong capability for leveraging relevant background information for response generation.

However, Gemini-pro's performance on verbal analogy questions was lower. Verbal analogy questions demand the model to grasp the relationship between word pairs and identify another pair with a similar connection. This task can be challenging for LLMs, as it necessitates not only understanding individual word meanings but also the intricate ways words can relate to each other. Interestingly, both ChatGPT-4 and

Section	Question Type	# of Wrong Answers (ChatGPT-4)	Gemini-pro (ChatGPT-4) Accuracy	# of Wrong Answers (ChatGPT-3.5-turbo)	Gemini-pro (ChatGPT-3.5-turbo) Accuracy
Quantitative	Math	128/370	30%	180/370	36%
	Geometry	55/142	**44%**	61/142	36%
	Algebra	168/374	27%	221/374	29%
	Statistics	37/128	19%	74/128	**43%**
Verbal	Reading Comprehension	46/199	**46%**	73/199	**56%**
	Sentence Completion	83/300	30%	168/300	38%
	Contextual Error	132/298	29%	176/298	34%
	Verbal Analogy	124/298	16%	197/298	26%
	Anomalous Word	116/300	30%	160/300	39%

Table 3: Gemini-pro effectiveness on incorrect answers from ChatGPT-4 and ChatGPT-3.5-turbo

ChatGPT-3.5-turbo also exhibited lower performance on this question type, potentially indicating a general limitation in current LLM technology.

It is worth noting that Gemini-pro showed strength in following instructions by responding with the answer only without explanation or additional context. Unlike ChatGPT-4 and ChatGPT-3.5-turbo, which occasionally included extraneous information in their responses, Gemini-pro consistently provided only the answer to the question, as instructed.

The result of this evaluation suggests that Gemini-pro is a promising LLM that shows particular strength in the Arabic language by being able to correctly answer a wide range of questions that were incorrectly answered by ChatGPT-4 and ChatGPT-3.5-turbo.

7 Conclusion and Future Work

This research paper introduces the Qiyas benchmark, a novel evaluation framework developed to comprehensively assess the mathematical reasoning and language understanding capabilities of large language models (LLMs) in the Arabic language. The Qiyas benchmark is a standardized General Aptitude Test (GAT) used for university admissions in Saudi Arabia, ensuring its quality and relevance to real-world assessment.

The key findings of this paper are:

1) ChatGPT-4 outperformed ChatGPT-3.5-turbo across both the quantitative (math) and verbal (language) sections of the benchmark. This suggests that the newer, more advanced model has made notable progress in Arabic language understanding and mathematical reasoning compared to its predecessor.
2) The performance of the models varied depending on the prompt setting (zero-shot, one-shot, 3-shot). In general, providing more contextual information through one-shot and 3-shot prompts improved the models' accuracy, particularly in the verbal section tasks like reading comprehension.
3) The results highlight the current limitations of state-of-the-art LLMs in handling the complexities of the Arabic language, including its unique morphology and writing system. This underlines the need for more Arabic-focused training data and model development efforts to enhance the mathematical and linguistic capabilities of future Arabic LLMs.

The release of the Qiyas benchmark represents a significant contribution to the field, as it provides a robust, standardized evaluation framework for assessing the capabilities of Arabic language models. This resource can drive the development of more capable Arabic LLMs by serving as a benchmark for progress and identifying specific areas requiring further research and improvement. Future work includes expanding the dataset to include image-based questions, enabling the evaluation of multimodal models' ability to integrate language and visual understanding for Arabic-based tasks. Additionally, assessing a wider range of state-of-the-art LLMs on the Qiyas benchmark will provide a more comprehensive understanding of the current capabilities and limitations of Arabic language AI systems. Overall, this work lays the foundation for advancing the state-of-the-art in Arabic language understanding and reasoning for large language models.

References

Alkaoud, M. (2024). A bilingual benchmark for evaluating large language models. *PeerJ Computer Science*, *10*, e1893. https://doi.org/10.7717/peerj-cs.1893

Almansor, E. H., Al-Ani, A., & Hussain, F. K. (2020). Transferring Informal Text in Arabic as Low Resource Languages: State-of-the-Art and Future Research Directions. In L. Barolli, F. K. Hussain, & M. Ikeda (Eds.), *Complex, Intelligent, and Software Intensive Systems* (pp. 176–187). Springer International Publishing. https://doi.org/10.1007/978-3-030-22354-0_17

Chow, J. C. L., Sanders, L., & Li, K. (2023). Impact of ChatGPT on medical chatbots as a disruptive technology. *Frontiers in Artificial Intelligence*, *6*. https://doi.org/10.3389/frai.2023.1166014

Education and Training Evaluation Commission. (n.d.). Education and Training Evaluation Commission. Retrieved April 1, 2024, from https://www.etec.gov.sa/en/qiyas

ETEC. (n.d.). *هيئة تقويم التعليم والتدريب—Media Center*. Eucation and Training Evaluation Commission | ETEC. Retrieved April 4, 2024, from https://spapi.etec.gov.sa:2443/ar/MediaAssets/aptbroch.pdf?csf=1&e=zc4F99

Feng, T. H., Denny, P., Wuensche, B., Luxton-Reilly, A., & Hooper, S. (2024). More Than Meets the AI: Evaluating the performance of GPT-4 on Computer Graphics assessment questions. *Proceedings of the 26th Australasian Computing Education Conference*, 182–191. https://doi.org/10.1145/3636243.3636263

Fowler, T., Pullen, S., & Birkett, L. (2023). Performance of ChatGPT and Bard on the official part 1 FRCOphth practice questions. *British Journal of Ophthalmology*. https://doi.org/10.1136/bjo-2023-324091

Geerling, W., Mateer, G. D., Wooten, J., & Damodaran, N. (2023). ChatGPT has Aced the Test of Understanding in College Economics: Now What? *The American Economist*, *68*(2), 233–245. https://doi.org/10.1177/05694345231169654

Gemini Team, Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., Lillicrap, T., Lazaridou, A., … Vinyals, O. (2024). *Gemini: A Family of Highly Capable Multimodal Models* (arXiv:2312.11805). arXiv. http://arxiv.org/abs/2312.11805

Giannos, P., & Delardas, O. (2023). Performance of ChatGPT on UK Standardized Admission Tests: Insights From the BMAT, TMUA, LNAT, and TSA Examinations. *JMIR Medical Education*, *9*(1), e47737. https://doi.org/10.2196/47737

Kaye, A. S. (2018). Arabic. In *The World's Major Languages* (3rd ed.). Routledge.

Nguyen, X.-H., Nguyen, H.-A., Cao, L., & Hana, T. (2023). *ChatGPT's Understanding of History A Comparison to Vietnamese Students and its Potential in History Education*. OSF. https://doi.org/10.35542/osf.io/8z9tj

OECD & The World Bank. (2022). *How Learning Continued during the COVID-19 Pandemic: Global Lessons from Initiatives to Support Learners and Teachers* (S. Vincent-Lancrin, C. Cobo Romaní, & F. Reimers, Eds.). OECD. https://doi.org/10.1787/bbeca162-en

OpenAI Platform. (n.d.). Retrieved April 4, 2024, from https://platform.openai.com

Temsah, O., Khan, S. A., Chaiah, Y., Senjab, A., Alhasan, K., Jamal, A., Aljamaan, F., Malki, K. H., Halwani, R., Al-Tawfiq, J. A., Temsah, M.-H., Al-Eyadhy, A., Temsah, O., Khan, S. A., Chaiah, Y., Senjab, A. M., Alhasan, K., Jamal, A., Aljamaan, F., … Al-Eyadhy, A. (2023). Overview of Early ChatGPT's Presence in Medical Literature: Insights From a Hybrid Literature Review by ChatGPT and Human Experts. *Cureus*, *15*(4). https://doi.org/10.7759/cureus.37281

Appendix A. Verbal Section Sample Questions

Question Type	Question	Translated Question
Reading Comprehension	الأسئلة التالية تتعلق بالنص الذي يسبقها، بعد كل سؤال أربع اختيارات، أحدها صحيح. المطلوب هو قراءة النص بعناية، واختيار الإجابة الصحيحة عن كل سؤال. *الحمية القاسية* *هناك من يمارس الحمية القاسية دون أن يكتسب العادات الغذائية الجيدة لذا يعود للسمنة مجدداً* يتحدث النص عن: أ) الحمية القاسية لا تؤدي لتخفيف الوزن ب) الحمية القاسية نادرا ما تنتهي بالفشل ج) الحمية تفيد الجسم حتى مع العادات الغذائية السيئة **د) غياب العادات الغذائية الجيدة يقلل من فائدة الحمية**	The following questions relate to the preceding text. After each question, there are four choices, one of which is correct. The task is to read the text carefully and select the correct answer for each question. *The Harsh Diet* *Some people follow a harsh diet without acquiring good eating habits, so they return to obesity again.* The text is talking about: A) Harsh diets do not lead to weight loss. B) Harsh diets rarely end in failure. C) Diet benefits the body even with bad eating habits. **D) Lack of good eating habits reduces the effectiveness of the diet**
Sentence Completion	تلي كل جملة من الجمل الآتية أربع اختيارات، أحدهما يكمل الفراغ أو الفراغات في الجملة إكمالاً صحيحاً. المطلوب هو اختيار الإجابة الصحيحة. *رمي .... في الطريق فعل غير ......* أ) الأقلام - تربوي ب) الأوراق - صحي ج) القاذورات - مُشين **د) المخلفات – حضاري**	Following each sentence are four choices, one of which completes the blank(s) in the sentence correctly. The task is to choose the correct answer. *Throwing .... on the road is a(n) .... act.* A) Pens - educational B) Papers - healthy C) Trash - shameful **D) Waste - uncivilized**
Contextual Error	لكل جملة مما يأتي أربع خيارات. المطلوب هو: تحديد الكلمة التي لا يتفق معناها مع المعنى العام للجملة. (الخطأ ليس إملائياً ولا نحوياً) *شخصية الفرد لها عدة صفات جسمية مثل حسن الهيئة وعقلية مثل التفكير ونفسية مثل الصدق والطول* أ) التفكير **ب) الطول** ج) جسمية د) شخصية	For each of the following sentences, there are four choices. The task is to identify the word that does not match its meaning with the overall meaning of the sentence. (The error is not related to spelling or grammar). *The individual's character has several physical traits like good physique, mental traits like thinking, and psychological traits like honesty and height.* A) Thinking **B) Height** C) Physical D) Character
Verbal Analogy	في بداية كل سؤال مما يأتي، كلمتان ترتبطان بعلاقة معينة، تتبعهما أربعة أزواج من الكلمات، أحدها ترتبط فيه الكلمتان بعلاقة مشابهة للعلاقة بين الكلمتين في بداية السؤال. المطلوب، هو: اختيار الإجابة الصحيحة *بناء:هدم* **أ) قبل:بعد** ب) ندم:حزن ج) نعاس:نوم د) هدوء:سكون	At the beginning of each question from the following, there are two words related in a certain way, followed by four pairs of words, one of which has a relationship similar to the relationship between the two words at the beginning of the question. The task is to choose the correct answer. *Construction: Destruction* **A) Before: After** B) Regret: Sadness C) Drowsiness: Sleep D) Quietness: Stillness

Appendix B. Quantitative Section Sample Questions

Question Type	Question	Translated Question
Math	إذا كان ثمن 6 دفاتر يعادل 12 ريال فكم يكون ثمن 9 دفاتر من نفس النوع؟ أ) ١٢ **ب) ١٨** ج) ٢٤ د) ٢٧	If the price of 6 notebooks is equivalent to 12 riyals, how much is the price of 9 notebooks of the same type? A) 12 **B) 18** C) 24 D) 27
Geometry	مستطيل محيطه 40 سم وطوله يزيد عن عرضه بمقدار 2 فما هي مساحته؟ **أ) ٩٩ سم** ب) ٤٠٠ سم ج) ٢٠ سم د) ١٠٨ سم	A rectangle has a perimeter of 40 cm, and its length is 2 times its width. What is its area? **A) 99 cm** B) 400 cm C) 20 cm D) 108 cm
Algebra	إذا كان س^2 - ص^2 = 16 وكان س - ص = 2 فإن س + ص = ... **أ) ٨** ب) ٢ ج) ٤ د) ١	If $x^2 - y^2 = 16$ and $x - y = 2$, then $x + y$ = ... **A) 8** B) 2 C) 4 D) 1
Statistics	عند رمي مكعب أرقام مرقم من 1 إلى 6 فما هو احتمال أن يكون الوجه العلوي عدد أولي؟ **أ) ٢/١** ب) ٣/١ ج) ٦/١ د) ٤/١	When throwing a cube numbered from 1 to 6, what is the probability that the top side is a prime number? **A) 1/2** B) 1/3 C) 1/6 D) 1/4

Analyzing Politeness in Arabic Tweets: A Preliminary Study

Hend Al-Khalifa[1*], Nadia GHEZAIEL HAMMOUDA[2*] and Maria BOUNNIT[3*]

[1]Information Technology Department, College of Computer and Information Sciences,
King Saud University, Riyadh, Saudi Arabia
[2]Department of Software Engineering, College of Computer Science and Engineering
University of Hail, Hail, Saudi Arabia
[3]College of Arabic Language, Cadi Ayyad University, Marrakesh, Morocco
[*]iWAN Research Group, King Saud University, Riyadh, Saudi Arabia

[1]hendk@ksu.edu.sa, [2]n.hamouda@uoh.edu.sa, [3]m.bounnit.ced@uca.ac.ma

Abstract

This paper explores the application of computational methods, specifically transfer learning and large language models (LLMs), for analyzing politeness in Arabic text. The study employs a subset of an Arabic dataset sourced from X (formerly Twitter), focusing on expressions in Modern Standard Arabic (MSA). The proposed approach involves fine-tuning pre-trained Arabic language models and applying zero-shot and few-shot learning methods using various LLMs. The results demonstrate the potential of these techniques for politeness analysis in Arabic social media content, with the fine-tuned models and LLMs achieving varying levels of performance across different evaluation metrics. The study highlights the need for further research to refine methodologies, expand datasets, and incorporate cultural nuances specific to Arabic communication contexts, particularly in the realm of social media interactions.

1 Introduction

Politeness is a fundamental aspect of human communication that helps maintain social harmony and facilitate effective interactions. In the Arabic language, politeness is deeply rooted in cultural norms and social structures, with various linguistic strategies employed to convey respect and courtesy (Ameri et al., 2023). With the increasing prevalence of online communication, particularly on social media platforms like Twitter (X), understanding and analyzing politeness in Arabic text has become a crucial area of research in natural language processing (NLP).

Recent advancements in NLP, such as transfer learning and large language models (LLMs), have shown promising results in various language understanding tasks. However, their application to Arabic politeness analysis remains largely unexplored. This paper aims to bridge this gap by investigating the effectiveness of transfer learning and LLMs in detecting politeness in Arabic social media text.

The main contributions of this paper are as follows:

1. We present a comparative study of transfer learning and LLMs for analyzing politeness in Arabic text, specifically focusing on social media content from Twitter (X). To the best of our knowledge, this is the first study to apply these techniques to Arabic politeness detection in the context of social media.
2. We evaluate the performance of fine-tuned Arabic language models (MarBERT and CamelBERT) and LLMs (GPT-4o-mini, Cohere Command, and JAIS 30B Chat) using various evaluation metrics, providing insights into their strengths and limitations for this task.

The rest of the paper is organized as follows: Section 2 provides a linguistic background on politeness in the Arabic language, discussing its cultural roots and various linguistic strategies. Section 3 reviews related work in computational politeness analysis and Arabic NLP. Section 4 describes the dataset used in this study and the annotation process. Section 5 outlines the experimental design, including the transfer learning and LLM approaches employed. Section 6 presents the results and discusses the performance of the different models. Section 7 conducts an error analysis to identify common challenges and limitations of the proposed approaches. Finally, Section 8 concludes the paper by summarizing the main findings and suggesting directions for future work.

2 Background

The concept of politeness has been extensively studied in pragmatics, particularly through the works of Brown and Levinson (Brown & Levinson, 1987). Their model introduces the idea of a face divided into positive and negative face needs. A positive face refers to the desire to be liked and appreciated, while a negative face involves the need for autonomy and freedom from imposition. Their strategies for maintaining face include:

- Bald on-record: Direct communication without any politeness strategies.
- Positive politeness: Strategies that emphasize camaraderie and mutual respect.
- Negative politeness: Strategies that emphasize the addressee's right to freedom and non-imposition.
- Off-record: Indirect communication that allows for plausible deniability.
- Withholding: Choosing not to say anything at all.

Robin Lakoff adds another dimension to this framework by emphasizing the importance of making the interlocutor feel good and joyful. Her principles of politeness include not imposing, giving options, and making the addressee feel good, highlighting the interpersonal and affective aspects of polite communication (Alaearji & Buraihi, 2021).

Politeness in the Arabic language is deeply rooted in cultural norms and social structures. The language employs various linguistic strategies to convey respect and maintain social harmony. These strategies may include:

- Forms of address: Using plural forms for singular addresses in formal situations (e.g., "هل يمكنكم" instead of "هل يمكنك"), and employing titles and honorifics (e.g., "دكتور", "أستاذ", "سعادة").
- Kinship terms: Using words like "عم" (uncle) or "خالة" (aunt) to address non-relatives respectfully in daily interactions.
- Teknonyms (كنية): Using "أبو" (father of) or "أم" (mother of) followed by a child's name as a form of respect.

As noted by (Ameri et al., 2023) many of these politeness features developed when Arab societies encountered other civilizations, particularly Persian culture. They also mentioned that politeness norms in Arabic are not static but evolve with societal changes.

3 Related Work

This section outlines the methods employed for computational politeness, which are broadly classified into statistical methods, deep learning-based approaches, and Large Language Models. While state-of-the-art research covers various languages, it is notable that politeness features have not been extensively explored in the context of Arabic.

Starting with statistical computational politeness, we found several studies that have utilized SVM methods. (Danescu-Niculescu-Mizil et al., 2013) developed two classifiers to predict politeness in English requests: a Bag of Words (BOW) classifier with unigram features and a linguistically informed classifier incorporating additional linguistic features. They used logistic regression on the SVM outputs to score politeness. However, (Hoffman et al., 2017) contested their approach, revalidating the politeness-labeling tool for broader application in social computing. (Kumar, 2021) worked on an SVM-based classifier for identifying politeness in Hindi. Kumar's approach includes classifiers using unigram and bigram features, as well as additional linguistic features like formulaic expressions and honorifics. (Fu et al., 2020) proposed a three-step pipeline: Plan, Delete, and Generate, for paraphrasing messages to adjust politeness. Their model uses Integer Linear Programming (ILP) to plan appropriate politeness strategies, deletes irrelevant markers, and integrates new strategies to produce a polite paraphrased message.

Recent advances in deep learning for computational politeness include various approaches for both identification and generation. For Politeness Identification, (Aubakirova & Bansal, 2016) used a Convolutional Neural Network (CNN) with the Stanford Politeness

Corpus to predict politeness, outperforming previous methods and identifying new politeness markers. Also, (Mishra et al., 2023) employed a hierarchical transformer network to accurately predict politeness by leveraging contextual information from previous dialogue utterances. In (Dasgupta et al., 2023), they proposed a graph-induced transformer network (GiTN) combining Graph Convolution Networks (GCN) and BERT for detecting formality and politeness in text.

Concerning the Politeness Generation, (Sennrich et al., 2016) introduced an attention-based encoder-decoder NMT system for translating English to German with controllable politeness via target-side T-V annotations. (Feely et al., 2019) proposed a formality-aware NMT system for English to Japanese translation that enhances translation quality, particularly for formal and polite sentences. (Niu & Bansal, 2018) developed three weakly supervised models—Fusion, Label-Fine-Tuning (LFT), and Polite Reinforcement Learning (Polite-RL)—for generating contextually consistent polite responses in open-domain dialogues without parallel data. (Firdaus et al., 2020), (Golchha et al., 2019), and (Wang et al., 2020) highlighted the importance of politeness in user satisfaction and retention. Golchha et al. proposed a reinforced pointer-generator model for courteous responses in customer-care dialogues. Firdaus et al. used a pointer-generator network to produce courteous responses in Hindi and English. While Wang et al. suggested a sequence-to-sequence framework to add politeness and positivity in customer support responses. (Mishra et al., 2022) introduced the Politeness Adaptive Dialogue System (PADS), which uses reinforcement learning to incorporate politeness into dialogue management based on user satisfaction feedback.

The last approach is based on Large Language Models (LLMs), such as GPT-3, LLaMA2, and ChatGPT, which are advanced generative AI systems with billions of parameters trained on extensive textual data. These models have significantly advanced various NLP tasks, including emotion recognition and dialogue. However, their ability to adhere to politeness norms remains a crucial issue. Research by (Li et al., 2023) and (Ziems et al., 2024) showed that state-of-the-art LLMs, like ChatGPT and GPT-3, perform reasonably well in predicting politeness and classifying utterances into polite, neutral, or impolite categories.

Despite significant advancements in computational politeness research across various languages, there remains a notable gap in addressing the unique linguistic and cultural aspects of politeness in Arabic. The intricate interplay between language, culture, and social norms in Arabic presents distinct challenges for computational models aiming to analyze and generate polite language. To bridge this gap, our paper focuses on analyzing politeness in Arabic, by applying transfer learning techniques.

4 Dataset

Our dataset consists of 500 tweets selected from the ArSarcasm database[1]. We specifically chose tweets labeled as "MSA" because it is considered formal, and we aimed to focus on the challenges presented by this linguistic variety. We also filtered the tweets based on the sentiment labeled as positive to examine whether a positively sentiment expression could help detect polite lexical markers more easily.

The tweets present a variety of themes and tones, ranging from political and religious subjects to cultural events and personal expressions. They are primarily written in MSA with occasional English phrases and dialectic words. The tweets presented entail a variety of subjects and moods from political or religious discussions to cultural events, and personal opinions. The Arabic employed in these tweets is split between MSA used in formal statements and quotes of religious nature on one hand and the vernacular languages used in less formal and more intimate interactions on the other. This dualism in language choices characterizes the richness and flexibility of Arabic which allows speakers to adjust their speech according to context as well as their audience.

In our work, we aim to detect politeness in the selected expressions. We have chosen to start with manual annotation, meaning we annotated the expressions without using available systems and software, adopting a linguistic approach. Based on the non-exhaustive analysis conducted by (Alaearji & Buraihi, 2021), we utilize the following tools:

- Politeness markers: A lexical tool examining the words used in each expression. If the expression contrains one or more words from

---

[1] https://github.com/iabufarha/ArSarcasm

the category of politeness, it is labeled as polite.

- Intention and Purpose of the Expression: A pragmatic tool arising from understanding the finality of the formulated expression. We opted for: appreciation, respect, love, admiration.

Tweet	politeness markers(lexical)	intention / finality (pragmatic)	Politeness?
خلق_جميل احترام اي انسان فقط #__ لأنه انسان مهما كانت ديانته	جميل / احترام	**Appreciation**	**POLITE**
"" القرار صهيوني"... #حماس "" ترفض منع استخدام مكبرات الصوت في المساجدhttps://t.co/jTAAch4M2w https://t.co/NjjE1gKjF4"	-	-	**NEUTRAL**
"الهلال الاحمر الكويتي»: وقعنا » اتفاقية مع «الهلال القطري» لتوزيع المساعدات الإغاثية على النازحين السوريين من حلب"	-	-	**NEUTRAL**
" لله درك #قطر وقادتها#قطر تلغي الاحتفال بـ #اليوم_الوطني تضامناً مع #حلب #سوريا #ستبقي_حلب... https://t.co/wV2pFqGjdD"	لله درك	**Blessing**	**POLITE**

Table 1: Examples of Annotated Tweets

We must nonetheless point out that our analysis is subject to a certain degree of subjectivity, considering the nature of the expressions analyzed (tweets), the absence of some elements that might be considered essential such as the knowledge of the speaker and the interlocutor, as well as the exact context and conditions of formulation. Additionally, some tweets may not be in pure MSA, which can affect the consistency of our analysis.

Table 1 shows sample tweets annotated based on their politeness markers and intention.

5 Experimental design

Our approach (as shown in Figure 1) is based on transfer learning, where we adapt finetuning pre-trained language models such as CamelBERT[2] and MarBERT[3] models to the task of politeness analysis using a dataset annotated with politeness labels. The process will be our baseline and it involves tokenizing the input text, formatting it appropriately, and adding task-specific layers to the pre-trained model, followed by training with adjusted hyper parameters to optimize performance. The model's effectiveness is evaluated using metrics like accuracy, precision, and F1-score on split datasets.

Additionally, we employed zero-shot and few-shot learning methods with GPT-4o-mini[4], Cohere Command[5] and JAIS 30B Chat[6] for politeness analysis, constructing prompts to guide the model and making predictions based on their pre-existing knowledge without fine-tuning. The selection of these models is justified by their unique strengths and relevance to the task. GPT-4o-mini's multimodal approach and superior non-English performance, Cohere Command efficiency and robustness, and JAIS 30B Chat's specific design for Arabic and English make them well-suited for a comprehensive evaluation of Arabic language processing capabilities. Note that Few-shot learning incorporates a small-annotated dataset (between 7 and 20 in our case) to provide context in the prompts.

---

[2] https://github.com/CAMeL-Lab/CAMeLBERT

[3] https://huggingface.co/UBC-NLP/MARBERT

[4] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[5] https://cohere.com/command

[6] https://www.core42.ai/jais.html

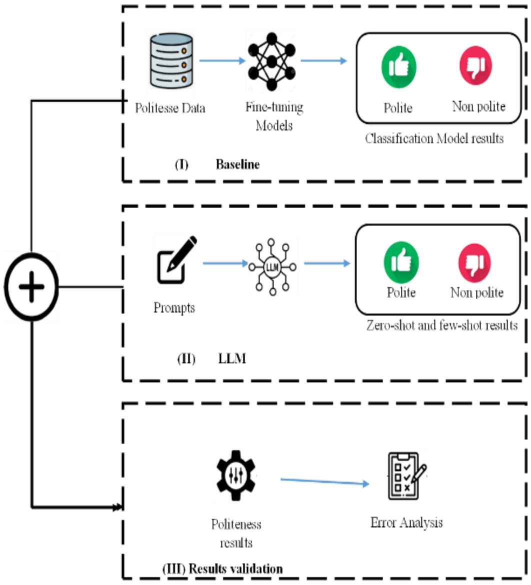


Figure 1: Overall summary of the proposed experiments.

6 Results and Discussion

In this section, we present the various outcomes of our experiments.

Model	Zero-Shot	Few-Shot (7 shots)	Few-Shot (20 shots)
GPT-4o-mini	65.13%	70.73%	66.39%
Cohere Command	54.31%	59.15%	58.87%
JAIS 30B Chat	**65.33%**	**70.87%**	**67.59%**

Table 3: Zero/Few shot learning Evaluation Results.

6.1 Baseline

Table 2 provides the results of MarBERT and CamelBERT models evaluated on accuracy, precision, and F1-score. MarBERT shows higher accuracy (54%) compared to CamelBERT (46%), indicating it classifies more instances correctly overall. However, CamelBERT excels in precision (37% vs. 27%) and F1-score (41.8% vs. 35%), demonstrating it is more reliable in positive predictions and better balanced between precision and recall. Overall, while MarBERT has better accuracy, CamelBERT offers superior performance in terms of precision and F1-score, suggesting it might be more effective for tasks where precision and recall are crucial.

Model	Accuracy	Precision	F1-score
MarBERT	0.54	0.27	0.35
CamelBERT	0.46	0.37	0.418

Table 2: Baseline Evaluation Results.

6.2 Zero/Few shot learning

Table 3 compares the performance of three language models: GPT-4o-mini, Cohere Command, and JAIS 30B Chat on the task of politeness classification in the Arabic language. The evaluation is conducted under three different settings: zero-shot learning, few-shot learning with 7 examples, and few-shot learning with 20 examples.

In the zero-shot setting, where the models are tested without any training examples, GPT-4o-mini and JAIS 30B Chat achieve similar accuracy scores of 65.13% and 65.33%, respectively. Cohere Command lags behind with an accuracy of 54.31%.

With few-shot learning using 7 examples, all three models show improvement. JAIS 30B Chat takes the lead with 70.87% accuracy, closely followed by GPT-4o-mini at 70.73%. Cohere Command also improves but remains in third place with 59.15% accuracy.

When the number of few-shot examples is increased to 20, there is a slight decrease in performance for all models compared to the 7-shot setting. JAIS 30B Chat maintains its lead with 67.59% accuracy, followed by GPT-4o-mini at 66.39% and Cohere Command at 58.87%.

Overall, JAIS 30B Chat and GPT-4o-mini demonstrate superior performance in both zero-shot and few-shot settings compared to Cohere Command for the task of Arabic politeness classification. The results also suggest that increasing the number of few-shot examples from 7 to 20 does not necessarily lead to improved performance for this particular task and dataset.

7 Error Analysis

The baseline models, MarBERT and CamelBERT, yielded divergent results in the politeness analysis of 500 manually annotated tweets. MarBERT classified all tweets as polite, indicating a biased output that exceeded linguistic explanations. In contrast, CamelBERT's predictions aligned well with the manual annotations, with only 12% of expressions remaining incompatible. This discrepancy can be attributed to two main factors: linguistic complexity and data bias. The complex structure and nuances of tweets, including ellipses, wordplay, and implicit references, pose a significant challenge for CamelBERT, requiring deep contextual understanding. Additionally, CamelBERT may have inherited and amplified biases present in its training data, particularly in the context of polite language in various contexts, including religious ones. This bias led to a loss of accuracy in predictions and limited the model's efficiency in politeness analysis.

On the other hand, the error analysis of the politeness prediction models, GPT-4o-Mini, Cohere Command, and JAIS, reveals several factors contributing to the discrepancies between manual annotations and model predictions. GPT-4o-Mini shows a binary approach to politeness, often missing the subtleties that might classify an expression as slightly polite or contextually polite, while Cohere Command exhibits a significant bias towards classifying tweets as polite, lacking a nuanced understanding of politeness gradations. JAIS performs better than the other models but still struggles with contextual understanding of polite content. The main issues identified include confusion between language and social practice, where the models fail to capture the intention and purpose of the entire sentence, as well as errors in analyzing certain sentences, possibly due to the presence of complex linguistic structures or ambiguous expressions. Additionally, the reliance on emojis, particularly heart-shaped ones, can disrupt the politeness prediction, as the models may interpret their presence as a sign of politeness, even when the rest of the tweet's content does not include lexical or pragmatic markers of politeness. These findings highlight the need for more sophisticated models that can better capture the nuances of politeness in Arabic text, taking into account both lexical markers and the overall intention and context of the communication.

8 Conclusion and Future work

In this paper, we presented a comparative study of transfer learning and large language models (LLMs) for analyzing politeness in Arabic text sourced from X (formerly Twitter). Our approach involved fine-tuning pre-trained Arabic language models, specifically MarBERT and CamelBERT, and applying zero-shot and few-shot learning methods using various LLMs, including GPT-4o-mini, Cohere Command, and JAIS 30B Chat. The results demonstrated the potential of these techniques for politeness analysis in Arabic social media content, with the fine-tuned models and LLMs achieving varying levels of performance across different evaluation metrics. The error analysis revealed several factors contributing to the discrepancies between manual annotations and model predictions, including confusion between language and social practice, errors in analyzing certain sentences, and reliance on emojis. Despite the challenges and limitations identified, this study highlights the potential of transfer learning and LLMs for analyzing politeness in Arabic text and underscores the importance of developing language-specific resources and incorporating cultural knowledge and pragmatic understanding into computational models.

Future research should focus on expanding the dataset, incorporating cultural and pragmatic knowledge, investigating advanced architectures, addressing data bias, and extending the insights gained from this study to other NLP tasks in Arabic. By addressing these research directions, future work can contribute to the development of more effective and culturally-aware politeness analysis models for Arabic text, ultimately leading to better communication and social interactions in the digital sphere.

Limitations

This study has several limitations that should be acknowledged. First, the dataset used in this research is relatively small, consisting of only 500 tweets, and focuses solely on Modern Standard Arabic (MSA). This limited dataset may not fully capture the diversity of Arabic dialects and the wide range of politeness expressions used in social media. Second, although the manual annotation was performed by a linguistic expert in the field, the process is still subject to a degree of subjectivity, as it relies on the annotator's understanding of politeness markers and pragmatic intentions, which may be influenced by their cultural background and individual perception. Third, the study focuses on a binary classification of politeness (polite or neutral), which may oversimplify the nuances of politeness in Arabic communication. Finally, the performance of the models may be affected by the limited size of the dataset and the potential biases present in the pre-trained language models used for transfer learning and few-shot learning.

References

Alaearji, A. A., & Buraihi, R. H. (2021). The Principle of Kindness and Politeness in Arabic Linguistic Thought. *International Journal on Humanities and Social Sciences*, *27*, 42–50. https://doi.org/10.33193/IJoHSS.27.2021.336

Ameri, M., Zeighami, A., & Mirahmadi, S. R. (2023). A study of the polite method in the Arabic language according to Brown and Levinson's theory of politeness. *Studies on Arabic Language and Literature*, *13 (36)*, 61–88. https://doi.org/10.22075/lasem.2022.23600.1286

Aubakirova, M., & Bansal, M. (2016). Interpreting Neural Networks to Improve Politeness Comprehension. In J. Su, K. Duh, & X. Carreras (Eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2035–2041). Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1216

Brown, P., & Levinson, S. C. (1987). *Politeness: Some Universals in Language Usage*. Cambridge University Press.

Danescu-Niculescu-Mizil, C., Sudhof, M., Jurafsky, D., Leskovec, J., & Potts, C. (2013). *A computational approach to politeness with application to social factors*. the 51st Annual Meeting of the Association for Computational Linguistics.

Dasgupta, T., Sinha, M., & Geetha Praveen, C. (2023). Graph Induced Transformer Network for Detection of Politeness and Formality in Text. *Companion Proceedings of the ACM Web Conference 2023*, 221–224. https://doi.org/10.1145/3543873.3587352

Feely, W., Hasler, E., & de Gispert, A. (2019). Controlling Japanese Honorifics in English-to-Japanese Neural Machine Translation. In T. Nakazawa, C. Ding, R. Dabre, A. Kunchukuttan, N. Doi, Y. Oda, O. Bojar, S. Parida, I. Goto, & H. Mino (Eds.), *Proceedings of the 6th Workshop on Asian Translation* (pp. 45–53). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5203

Firdaus, M., Ekbal, A., & Bhattacharyya, P. (2020). Incorporating Politeness across Languages in Customer Care Responses: Towards building a Multi-lingual Empathetic Dialogue Agent. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 4172–4182). European Language Resources Association. https://aclanthology.org/2020.lrec-1.514

Fu, L., Fussell, S., & Danescu-Niculescu-Mizil, C. (2020). Facilitating the Communication of Politeness through Fine-Grained Paraphrasing. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 5127–5140). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.416

Golchha, H., Firdaus, M., Ekbal, A., & Bhattacharyya, P. (2019). Courteously Yours: Inducing courteous behavior in Customer Care responses using Reinforced Pointer Generator Network. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 851–860). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1091

Hoffman, E. R., McDonald, D. W., & Zachry, M. (2017). Evaluating a Computational Approach to Labeling Politeness: Challenges for the Application of Machine Classification to Social Computing Data. *Proc. ACM Hum.-Comput. Interact.*, *1*(CSCW), 52:1-52:14. https://doi.org/10.1145/3134687

Kumar, R. (2021). *Towards automatic identification of linguistic politeness in Hindi texts* (arXiv:2111.15268). arXiv. https://doi.org/10.48550/arXiv.2111.15268

Li, C., Pang, B., Wang, W., Hu, L., Gordon, M., Marinova, D., Balducci, B., & Shang, Y. (2023). How Well Can Language Models Understand Politeness? *2023 IEEE Conference on Artificial Intelligence (CAI)*, 230–231. https://doi.org/10.1109/CAI54212.2023.00106

Mishra, K., Firdaus, M., & Ekbal, A. (2022). Please be polite: Towards building a politeness adaptive dialogue system for goal-oriented conversations. *Neurocomputing*, *494*, 242–254. https://doi.org/10.1016/j.neucom.2022.04.029

Mishra, K., Firdaus, M., & Ekbal, A. (2023). Predicting Politeness Variations in Goal-Oriented Conversations. *IEEE Transactions on Computational Social Systems*, *10*(3), 1095–1104. IEEE Transactions on Computational Social Systems. https://doi.org/10.1109/TCSS.2022.3156580

Niu, T., & Bansal, M. (2018). Polite Dialogue Generation Without Parallel Data. *Transactions of the Association for Computational Linguistics*, *6*, 373–389. https://doi.org/10.1162/tacl_a_00027

Sennrich, R., Haddow, B., & Birch, A. (2016). Controlling Politeness in Neural Machine Translation via Side Constraints. In K. Knight, A. Nenkova, & O. Rambow (Eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 35–40). Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1005

Wang, Y.-C., Papangelis, A., Wang, R., Feizollahi, Z., Tur, G., & Kraut, R. (2020). *Can You be More Social? Injecting Politeness and Positivity into Task-Oriented Conversational Agents* (arXiv:2012.14653). arXiv. https://doi.org/10.48550/arXiv.2012.14653

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can Large Language Models Transform Computational Social Science? *Computational Linguistics*, *50*(1), 237–291. https://doi.org/10.1162/coli_a_00502

FeruzaSpeech: A 60 Hour Uzbek Read Speech Corpus with Punctuation, Casing, and Context

Anna Povey
Redmond High School, WA, USA
`annapovey@gmail.com`

Katherine Povey
University of Washington, USA
`katherinepovey@gmail.com`

Abstract

This paper introduces FeruzaSpeech, a read speech corpus of the Uzbek language, containing transcripts in both Cyrillic and Latin alphabets, freely available for academic research purposes. This corpus includes 60 hours of high-quality recordings from a single native female speaker from Tashkent, Uzbekistan. These recordings consist of short excerpts from a book and BBC News. This paper discusses the enhancement of the Word Error Rates (WERs) on CommonVoice 16.1's Uzbek data, Uzbek Speech Corpus data, and FeruzaSpeech data upon integrating FeruzaSpeech.

1 Introduction

The Uzbek language, the official language of Uzbekistan, boasts upwards of 31 million native speakers across Central Asia. [1]Advancement in neural network models and deep learning have significantly improved automated speech recognition (ASR) and text-to-speech (TTS) technology in recent decades. Large freely available English datasets, such as LibriSpeech (Panayotov et al., 2015), Libriheavy (Kang et al., 2023b), and GigaSpeech (Chen et al., 2021) are now more robust then ever, however, datasets for training these models in Uzbek are scarce.

In January 2023, the Uzbek government fully transitioned from using the Cyrillic alphabet to using the Latin alphabet [2], yet the country continues to use both alphabets. FeruzaSpeech is the first dataset to offer both Cyrillic and Latin transcription. To the best of our knowledge, FeruzaSpeech is also the only corpus to provide Cyrillic transcriptions at all. Datasets originally using Latin transcription cannot yet be accurately converted into Cyrillic text using online conversion calculators because there are a few discontinuities between the two alphabets. An example is when conversion calculators like this one[3] are used on Cyrillic text that include the soft sign ь, it is either lost or can be incorrectly reproduced becoming a hard sign ъ.

Table 1: Conversion Calculator on Польша

Cyrillic to Latin	Latin to Cyrillic
Польша -> Pol'sha	Pol'sha -> Полъша

FeruzaSpeech aims to promote the development of speech recognition and speech synthesis technologies for the use of Uzbek speakers. Because this is a single speaker dataset with an absence of environmental noise it is better for STT when used in addition to other available speech corpuses. The dataset may be suitable for TTS applications, but such experiments are beyond the scope of this paper. It complements existing ASR datasets such Uzbek Speech Corpus (USC) (Musaev et al., 2021), consisting of 105 hours from 958 speakers, and the Common Voice Uzbek Dataset (Ardila et al., 2019) [4], with 265 hours from over 2,000 speakers. We chose these two corpuses because they were the only two other published datasets. When combined with these datasets, FeruzaSpeech enhances ASR model training.

2 FeruzaSpeech Corpus

This section describes the layout of the FeruzaSpeech corpus metadata, transcription, and audio format. Instructions for downloading and utilizing the data can be found on HuggingFace. [5]

2.1 Dataset Type

FeruzaSpeech consists of audio-book recordings from the texts of the book Choliqushi, a classic romance novel, and BBC Uzbek News read by our

---

[1] `https://www.worlddata.info/languages/uzbek.php`

[2] `https://interfax.az/view/826747`

[3] `https://uzlatin.com/`

[4] Download Page: `https://commonvoice.mozilla.org/en/datasets`

[5] `https://huggingface.co/datasets/k2speech/FeruzaSpeech`

voice actress, Feruza. Table 2 shows the duration of each type of recording within the dataset. Initially read in the Cyrillic alphabet, the texts were converted to Latin using online tools[6], with some grammatical errors being manually fixed after the use of the conversion calculator. The final transcription provides Uzbek text in both the Cyrillic and Latin alphabets.

Table 2: FeruzaSpeech Recordings

Type	Total
Book	21.57h
BBC Uzbek	38.04h
Total	59.61h

2.2 Evaluation and Training Sets

FeruzaSpeech includes "Dev" (development), "Test" (testing), and "Train" (training) sets as detailed in Table 3. Both the Dev and Test sets only include BBC articles, while the Train set also includes the Choliqushi novel.

Table 3: FeruzaSpeech Sets

Sets	Total
Dev	2.93h
Test	4.08h
Train	52.09h

2.3 Audio Format

The corpus contains high-quality, single-channel, 16-bit .wav audio files, available in 16kHz for ASR. The average recording length is 16.39 seconds, the minimum length is 3.78 seconds, and the maximum length is 50.69 seconds. Our segments are recordings of one to two full sentence and are much longer than the segments of USC (Musaev et al., 2021), that are mostly 2 to 3 seconds.

2.4 Sample Text

Table 4 shows example excerpts from the CommonVoice and USC datasets in comparison to our proposed FeruzaSpeech dataset.

We can see that CommonVoice text was normalized but it has punctuation and casing, USC text is normalized to remove casing and punctuation, and FeruzaSpeech retains casing and punctuation. Regarding the choice to not normalize casing or punctuation, deep learning models have recently become powerful enough that for tasks like ASR and TTS it is now feasible to use "natural" text with no normalization. For instance, the recent E2TTS text-to-speech system (Eskimez et al., 2024) is trained on data from Libriheavy (Kang et al., 2024) which is completely un-normalized. The use of un-normalized text for training tends to simplify speech processing systems because it could avoid the need for text normalization and inverse text normalization modules. Table 5 shows how the transcripts were provided in Latin and Cyrillic, but within this paper we only used Latin transcripts for comparison with available Latin datasets.

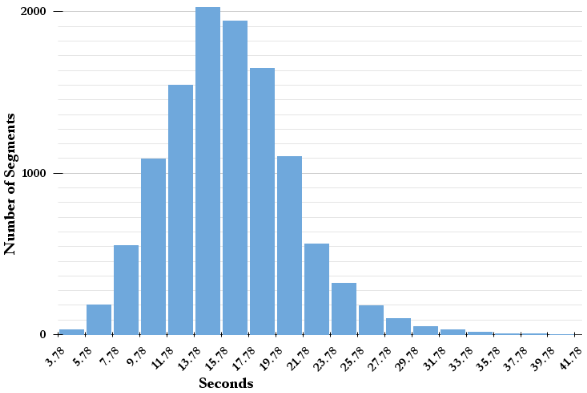


Figure 1: Length of FeruzaSpeech Segments

Table 4: One Sentence of Sample Text from each of the Three Datasets with English Translation That Is Not in the Dataset for Reader

CommonVoice	USC	FeruzaSpeech
— Non dema! — dedi. — Nonni otini atama!	shundan so’ng u sen aytmasang men aytaman degandek qaradi	20 iyul kuni O‘zbekistonda 562 holatda kasallik qayd etilgan.
— Don’t say bread! — he said. — Don’t utter the word bread!	after that he looked like if you don’t tell I will	On July 20, 562 cases of the disease were recorded in Uzbekistan.

---

[6] `https://www.lexilogos.com/keyboard/uzbek_conversion.htm` and `https://uzlatin.com/`

Table 5: FeruzaSpeech Excerpt in Latin and Cyrillic with English Translation That Is Not in the Dataset for Reader

FeruzaSpeech Latin	**FeruzaSpeech Cyrillic**
Ayni damda ishlatilib turilgan biometrik pasportlar 2019 yil 1 yanvardan deyarli yaroqsiz holatga keladi.	Айни дамда ишлатилиб турилган биометрик паспортлар 2019 йил 1 январдан деярли яроқсиз ҳолатга келади.
Biometric passports, which are currently in use, will become almost useless from January 1, 2019.	

3 Experiments

To build our models we utilized the Next-Gen Kaldi framework and followed two recipes; the Icefall CommonVoice Stateless RNN-T Conformer model taken from the pruned_transducer_stateless7 recipe[7] and the Librispeech zipformer[8], noting significant improvements in WER. These two models were selected because Icefall already contains CommonVoice scripts using pruned_transducer_stateless7 in the French language and Librispeech zipformer is the current state of the art model in Next-Gen Kaldi. In our experiments we utilized three datasets: Common Voice 16.1 (CV), FeruzaSpeech (FS), and Uzbek Speech Corpus (USC). All models were trained for 60 epochs. Table 6 outlines the duration of each training dataset.

Table 6: Training Dataset Duration

Datasets	**Training Duration**
CV	54.88h
FS	52.09h
USC	90.70h
CV + FS + USC	197.68h

3.1 Pruned-transducer-stateless7 Model

The Stateless RNN-T Conformer model (Kang et al., 2023a) is a stateless transducer (Gulati et al., 2020) with a conformer encoder that reduces memory consumption, and it outperformed the small zipformer model for every test set. All models in Tables 7 and 8 are trained with Casing and Punctuation. Table 7 presents the WERs when the model is scored with Casing and Punctuation, while Table 8 presents the WERs when the model is scored with Uppercase No Punctuation.

Table 7: The WERs of **Stateless RNN-T Conformer model** scored with **Casing and Punctuation (C&P)** Common Voice 16.1 (CV), Uzbek Speech Corpus (USC), FeruzaSpeech (FS)

Method	**Dataset**	**cv-test**	**fs-test**	**usc-test**
greedy search	CV	33.95	32.9	51.07
	FS	89.54	11.58	85.67
	CV+FS	32.49	9.93	46.89
	CV+FS+USC	29.91	9.79	12.05
modified beam search	CV	31.98	31.88	51.61
	FS	89.10	11.25	85.22
	CV+FS	30.47	9.85	48.6
	CV+FS+USC	27.81	9.56	11.67

Table 8: The WERs of **Stateless RNN-T Conformer** model scored with **Uppercase No Punctuation (UNP)** Common Voice 16.1 (CV), Uzbek Speech Corpus (USC), and FeruzaSpeech (FS)

Method	**Dataset**	**cv-test**	**fs-test**	**usc-test**
greedy search	CV	21.03	20.15	35.11
	FS	87.18	5.85	77.78
	CV+FS	18.91	4.44	30.53
	CV+FS+USC	12.07	4.17	12.05
modified beam search	CV	20.16	19.34	34.03
	FS	86.26	5.50	76.24
	CV+FS	18.33	4.24	29.67
	CV+FS+USC	11.17	4.05	11.67

3.2 Zipformer Model

We followed a similar procedure for the zipformer model as we did for the Stateless RNN-T Conformer model. This time, we trained a separate model on each of the following three datasets: CV, FS, CV+FS. This differs from the previous section because we excluded the USC training set. Once again, we recorded the WER for each model when tested on each of the following test sets: cv-test, fs-test, and usc- test, sharing results for both the greedy search and modified beam search as methods of decoding. All models in Table 9 and 10 are trained with Casing and Punctuation. Table 9 presents the WERs when the model is scored with Casing and Punctuation, while Table 10 presents the WERs when the model is scored with Uppercase No Punctuation.

---

[7] `https://github.com/k2-fsa/icefall/tree/master/egs/commonvoice/ASR/pruned_transducer_stateless7`

[8] `https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/zipformer`

Note that the Common Voice recipe with default settings in the Icefall project wasn't converging for zipformer (Kang et al., 2023b), so we used "small zipformer" [9] (Yao et al., 2024) parameters to account for the size of our datasets.

Table 9: The WERs of **zipformer** model scored with **Casing and Punctuation (C&P)** Common Voice 16.1 (CV) and FeruzaSpeech (FS)

Method	Dataset	cv-test	fs-test	usc-test
greedy search	CV	37.00	34.54	53.4
	FS	93.09	14.32	N/A
	CV+FS	35.90	11.05	52.86
modified beam search	CV	33.96	32.41	54.07
	FS	92.61	13.28	N/A
	CV+FS	33.15	10.75	53.08

Table 10: The WERs of **zipformer** model scored with **Uppercase No Punctuation (UNP)** Common Voice 16.1 (CV) and FeruzaSpeech (FS)

Method	Dataset	cv-test	fs-test	usc-test
greedy search	CV	23.01	20.94	38.75
	FS	91.34	8.97	N/A
	CV+FS	22.34	5.45	37.4
modified beam search	CV	21.92	20.21	37.44
	FS	90.54	7.88	N/A
	CV+FS	21.35	5.10	35.94

4 Results

When adding the FeruzaSpeech dataset to the CommonVoice16.1 dataset while training the Stateless RNN-T Conformer model, WER improved 1.49 to 2.12 percent absolutely on cv-test and 3.01 to 4.58 percent absolutely on usc-test in Tables 7 and 8. And for the Zipformer model, WER improved 0.57 to 1.1 percent absolutely on cv-test and 0.54 to 1.5 percent absolutely on usc-test in tables 9 and 10. This shows that FeruzaSpeech contains quality data and is a useful addition to the current public library of Uzbek speech corpuses for TTS applications. Also, the paper presenting the USC dataset (Musaev et al., 2021) reports that the usc-test had a WER of 17.4%. Our best result for the usc-test WER is 11.67%, which is an improvement of 5.73%. According to Table 6 and 7, when a Stateless RNN-T Conformer model was built using all three datasets combined: CV, FS, and USC, and using modified beam search as the decoding method, the model produced the best WERs for every test. Our best recorded WER on the Common Voice test set is 11.17%, as shown in Table 7. The best WER for the FeruzaSpeech test set is 4.05%, and the best WER for the Uzbek Speech Corpus test is 11.67%.

5 Conclusion

The development of FeruzaSpeech is a significant step forward in the field of Uzbek speech technology. By offering a dual alphabet corpus, this project bridges the gap between the use of Cyrillic and Latin scripts for Uzbek speakers. Our work also highlights the need for accurate alphabet conversion tools, specifically for more nuanced aspects of the language such as the soft sign (ь), which tends to be lost in translation from Cyrillic to Latin.

Through integrating FeruzaSpeech with existing Uzbek datasets, notable improvements in WERs were demonstrated. In the future, we will provide this same data in a higher sampling rate and bit depth that will be more suitable for TTS. Since we recognize the value of continuity in voice data for TTS applications, our future endeavors will also focus on expanding this corpus with additional recordings from the same native speaker. This strategy aims to enrich the dataset with consistent voice quality and style across the corpus which is essential for developing TTS models.

In sum, FeruzaSpeech is beneficial for ASR model enhancement when used in addition to existing Uzbek language datasets, as observed in WER improvements. Applications of this dataset for TTS will also be explored.

6 Limitations

FeruzaSpeech is not an effective stand alone corpus for STT applications and should be used in compliment with other corpuses such as the Common Voice Uzbek Dataset and Uzbek Speech Corpus explored above. FeruzaSpeech has an average segment length of 16.39 seconds which each contain one or two full sentences which could be segmented into shorter utterances. The audio has no background noise and contains a singular female speaker which is not optimal for STT.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-

---

[9] `https://github.com/k2-fsa/icefall/blob/master/egs/librispeech/ASR/RESULTS.md`

multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition.

Wei Kang, Liyong Guo, Fangjun Kuang, Long Lin, Mingshuang Luo, Zengwei Yao, Xiaoyu Yang, Piotr Żelasko, and Daniel Povey. 2023a. Fast and parallel decoding for transducer. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2023b. Libriheavy: a 50,000 hours asr corpus with punctuation casing and context. *arXiv preprint arXiv:2309.08105*.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2024. Libriheavy: a 50,000 hours asr corpus with punctuation casing and context.

Muhammadjon Musaev, Saida Mussakhojayeva, Ilyos Khujayorov, Yerbolat Khassanov, Mannon Ochilov, and Huseyin Atakan Varol. 2021. Usc: An open-source uzbek speech corpus and initial speech recognition experiments. In *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*, pages 437–447. Springer.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition.

Bulgarian Grammar Error Correction with Data Augmentation and Machine Translation Techniques

Bozhidar Klouchek
The University of Manchester
`klouchek.bozhidar@gmail.com`

Riza Batista-Navarro
The University of Manchester
`riza.batista@manchester.ac.uk`

Abstract

Grammar Error Correction (GEC) in Bulgarian is particularly difficult because of the lack of specialised linguistic resources and the highly inflected nature of the language. To facilitate research in the field we release three datasets created using data augmentation techniques inspired from phonetic and syntactic phenomena in the language. The datasets include a comprehensive machine-readable dictionary and two error correction datasets containing examples of spelling and grammar mistakes, respectively. Additionally, we employed an encoder-decoder transformer architecture, specifically multilingual T5 (mT5), to address the task of GEC using Neural Machine Translation (NMT). The proposed fine-tuned model outperforms contemporary large language model (LLM)-based solutions such as GPT and BgGPT by scoring an F0.5-score of 68.18%. It is also the most preferable in terms of output readability and correctness according to the human-based evaluation we conducted.

1 Introduction

Bulgarian is a highly inflected language, i.e., words take on different forms to convey information relating to gender, number, article, tense, case and other properties. Because of this, a slight syntactic difference can drastically change a word's meaning. For example, the word 'approach' in Bulgarian ('доближавам') has 51 different forms (Simov et al., 2004), all carrying different nuances about the speaker, the action's time frame and the level of emotion used in uttering the word. This, along with Bulgarian's intricate grammar, spelling and punctuation, makes mastering the language a unique challenge.

Despite the fact that as of 2011, the literacy rate in Bulgaria is 98.7% (National Statistical Institute, 2023), the language literacy performance of young Bulgarian students learning the language is lower than the average for tested countries (OECD, 2023), placing Bulgaria's mean score significantly below the average. This is troubling, as literacy is the foundation of language proficiency and is crucial for employability (Suarta et al., 2017), education (Castro et al., 2011) and social integration (Abdul-Rahaman et al., 2023).

Bulgarian Natural Language Processing (NLP) researchers could help alleviate this problem by creating: (a) linguistic resources, i.e., datasets, that facilitate the development of language literacy tools, and (b) error correction tools for Bulgarian. However, such resources and tools are currently lacking.

Datasets that could potentially facilitate the development of Bulgarian NLP tools include machine-readable dictionaries and error correction data. An official Bulgarian dictionary called the Institute for Bulgarian Language Online Dictionary[1] has been made publicly available. However, it comes with shortcomings that make it unsuitable for supporting the development of NLP tools. These include its inability to recognise words unless they are in their base word form, low confidence in recognising words that do not have a Bulgarian etymological origin, and lack of access to the entire word collection (preventing download by bulk). Meanwhile, error correction data is often required by systems that are developed or trained to assess language validity. This pertains to datasets that contain examples of spelling and grammar mistakes. There are many datasets of this kind for high-resource languages, e.g. English (Geertzen et al., 2013; Granger et al., 2009) and German (Meisel, 2020). However, no such Bulgarian resource is available.

Error correction tools are underpinned by models that verify a text's linguistic validity, focussing on different aspects of the language, including punctuation, spelling and grammar. Bulgarian spelling correction has been explored using noisy text mod-

---

[1] `https://ibl.bas.bg/rbe/lang/en/`

els (Gerdjikov et al., 2013). This approach was developed using the IMPACT BG dataset[2] which consists of 19th century Bulgarian newspaper articles, which are not indicative of modern Bulgarian communication. Grammar Error Correction (GEC) research for Bulgarian is scarce, most likely because of the short supply of error correction data and the inflectional nature of the language.

In this paper we aim to address the research gap caused by the lack of appropriate linguistic resources and absence of established solutions for error correction by releasing open-source datasets supporting language literacy and training a generative model for automatic GEC. We hope that this will encourage other members of the research community to build and compare solutions for Bulgarian language literacy tasks.

Our contributions include: (1) the creation of three datasets including a machine-readable Bulgarian dictionary that includes word inflections and Part-of-Speech tags, and two datasets produced using data augmentation, both of which contain pairs of erroneous and corrected sentences, one with spelling errors and the other with grammatical mistakes; and (2) the development and evaluation of a GEC solution based on fine-tuning a multilingual T5 (mT5) model (Xue et al., 2021) for neural machine translation of erroneous to correct text. The model,[3] along with the dictionary,[4] spelling error[5] and grammar error[6] datasets, are all open-source and available for public use.

2 Related Work

Dataset creation and error correction techniques are both pivotal for the success of automated language literacy tools. We review how these have been approached for Bulgarian and similar highly inflected languages.

2.1 Bulgarian Datasets

Below, we provide an overview of previously reported work on the development of Bulgarian linguistic resources and error correction datasets.

Linguistic Resources. Open-source Bulgarian linguistic resources have been published from as early as two decades ago. Among these is the BulTreeBank project (Simov et al., 2002), developed by the Bulgarian Academy of Sciences (BAS), which is considered to be the first successful initiative for large-scale curation for linguistic resources in Bulgarian NLP. The CLaRK system (Simov et al., 2003) is another notable achievement, presenting an automated system for corpora development that includes part-of-speech (POS) tagging and dependency extraction, utilising regular cascaded grammars. The CLaRK system is particularly useful for high-quality POS tagging in Bulgarian; we employed it in our work to identify candidates that can form the basis for inducing artificial errors.

More recent projects developed textual corpora that were drawn from specific domains such as law and medicine (Koeva et al., 2020; Boytcheva et al., 2020), as well as speech corpora (Dimitrova, 2021). However, no established Bulgarian error correction datasets have been released, hindering the progress of NLP researchers on error correction tasks.

Error Correction Datasets. Error correction data is particularly difficult to come by, as it necessitates a sophisticated approach to collection and/or generation of erroneous use of language. Systems using high-resource languages, like English (Dolgova and Mueller, 2019) and Chinese (Rao et al., 2018) rely on authentic *learner data* created by learners of the language, which can be then annotated manually. Low-resource language systems, however, tend to use *synthetic data* generated through *data augmentation*. This technique does not require language learners, rather, it generates the error correction data automatically by either:

- round-trip translation from error-free text, resulting in ungrammatical sentences (Lichtarge et al., 2019), or
- directly inducing errors in error-free text (Grundkiewicz and Junczys-Dowmunt, 2019; Lee and Seneff, 2008; Izumi et al., 2003).

Data augmentation was proven to be especially useful for low-resource languages (Solyman et al., 2023), as it provides a sustainable solution to the data scarcity problem. In this project, we chose to create artificial erroneous data by directly inducing grammatical errors based on predetermined linguistic rules, as there is a distinct lack of publicly

---

[2] `https://www.digitisation.eu/datasets/impact-language-resources/`

[3] `https://huggingface.co/thebogko/mt5-finetuned-bulgarian-grammar-mistakes`

[4] `https://huggingface.co/datasets/thebogko/bulgarian-dictionary-2024`

[5] `https://huggingface.co/datasets/thebogko/bulgarian-spelling-mistakes`

[6] `https://huggingface.co/datasets/thebogko/bulgarian-grammar-mistakes`

accessible learner data from humans.

2.2 Grammar Error Correction (GEC)

GEC approaches focus on transforming erroneous text to its correct version by identifying mistakes and recommending suggestions. These are typically based on machine translation (MT) methods.

Machine Translation Models. Statistical Machine Translation (SMT) is a probabilistic approach applied to GEC which, given an erroneous text sequence $e_1, e_2, \ldots e_m$, identifies the corrected text sequence $c_1, c_2, \ldots c_n$ that maximises the probability $p(c_1, c_2, \ldots c_n | e_1, e_2, \ldots e_m)$. This approach is often supported by a language model (Wang et al., 2021), ensuring that the corrections are fluent. The first error correction work based on SMT focussed on noun errors (Brockett et al., 2006).

Neural Machine Translation (NMT) differs from SMT in that it utilises *neural networks* to generate corrected text output (target) given erroneous input (source). Its strength lies in the ability of neural networks to generalise, allowing NMT systems to perform much better than SMT in correcting unseen error types (Wang et al., 2021). The first time it was used for GEC (Yuan and Briscoe, 2016) was ten years after SMT was first attempted, becoming the predominant approach to solving the error correction task. Diverse architectures have been used in NMT, such as recurrent neural networks (RNNs) (Yuan and Briscoe, 2016), convolutional neural networks (CNNs) (Chollampatt and Ng, 2018; Solyman et al., 2019) and transformers (Zhao et al., 2019; Grundkiewicz et al., 2019). Because of the success of NMT approaches we utilise it for our GEC task.

GEC for Highly Inflected Languages. GEC research for Bulgarian is scarce, likely because of the short supply of error correction data and the inflectional nature of the language. Some efforts have been made to detect noun-adjective disagreement (Borisova et al., 2014) and to investigate how grammars can be used for error correction (Kubon and Plátek, 1994), but at the time of writing no machine translation approach has been proposed for GEC in Bulgarian.

Our work aims to remedy this by training a transformer-based model on a large collection of grammar error mistakes and their corresponding corrections, motivated by studies showing that MT-based error correction systems for morphologically rich languages require large amounts of training data (Rozovskaya and Roth, 2019). We employed the mT5 model, given that it obtained encouraging results in the correction of highly inflected languages like Ukrainian (Lytvyn et al., 2023).

3 Creating Language Literacy Datasets

One of our objectives is the development of high-quality and open-source datasets that can be used for a diverse range of tasks that support language literacy. We showcase a comprehensive dictionary and two error correction datasets consisting of sentence pairs: one dataset contains spelling errors and the other contains grammatical mistakes.

3.1 A Machine-Readable Dictionary

Our Bulgarian dictionary contains 1,147,600 entries, each with a term and a corresponding part-of-speech (POS) tag. In this project's context, a *term* is defined as either the base form (lemma) of a word or an inflected form; in both cases, we only include single-word terms. This would allow a spell-checking system to perform a simple check for each token from a user's input to determine its validity.

Data Collection and Preprocessing. We firstly collected entries from two major open-source collections[7,8] due to their popularity and sufficient word coverage. It is worth noting that some inflected Bulgarian words, particularly verbs, can be supported by particles. For example, the word 'ям' ('eat') can change to 'ял' in some forms depending on the tense.

- 'щях да съм ял'
- 'бил съм ял'
- 'ял си'

Since the aforementioned dictionaries include these multi-word terms, while a spell-checking system would be expected to judge single tokens' validity on their own, we break up these multi-word terms and only look at unique sequences of characters. In this way we significantly decrease the number of terms, while still maintaining the dictionary's ability to determine if a word is spelled correctly.

Labelling. Additionally, the POS tag of each term is carried over from the sources we used. The tagging scheme includes 11 tags and is based on BulTreeBank's tagging scheme (Simov et al.,

---

[7] `https://slovored.com`

[8] `https://rechnik.chitanka.info`

2004), but was simplified by including only a single capitalised letter for the high-level role of the tag (e.g., `Amsf`, `Ansd`, etc. all conflate to `A` for ‘adjective’). Including these tags allows the dictionary to differentiate between homonyms. For example, the word ‘син’ describes both the adjective ‘blue’ and the noun ‘son’, so both are included in the dictionary with different POS tags.

3.2 Error Correction Datasets

Because of the scarcity of learner data in Bulgarian, we propose to collect Bulgarian text data and automatically induce spelling and grammar mistakes using data augmentation techniques. This approach allows us to generate pairs of correct-erroneous sentences, which will be necessary for training models to correct mistakes.

3.2.1 Error-inducement Algorithm

Not all errors can appear in all sentences, as they have specific phonetic, grammatical or lexical requirements. We defined an algorithm for inducing errors that takes a collection of source correct sentences C and a collection of error types T and returns a collection of unique tuples P, each tuple including three elements: a correct sentence $c \in C$, an erroneous sentence e and an error type $t \in T$.

3.2.2 Dataset for Spelling Error Correction

Our spelling error dataset consists of 23,719 pairs of Bulgarian sentences. In each pair, one sentence is the original sentence collected from the source corpus, which is presumed to be correct. The second one is an erroneous version of the correct one, including 1-3 spelling errors of the same type. The dataset spans 7 different error classes based on different linguistic phenomena in Bulgarian and each pair is labelled with one of those classes. To produce this dataset, the steps described below were carried out.

Data Collection and Preprocessing. The source data used to generate this dataset is Bulgarian Wikipedia articles, as we consider the quality of text in Wikipedia as being sufficient for our purposes. Overall, 28 Wikipedia articles were collected. The articles were fed into an spaCy implementation of a preprocessing pipeline specifically for Bulgarian text (Berbatova and Ivanov, 2023). Specifically, the articles underwent sentence segmentation, tokenisation and POS tagging.

To remove noisy sentences, two filters are applied, removing any sentences with fewer than three words or those without any verbs. This eliminated any sentences which are too short to be useful erroneous candidates. A total of 5817 sentences were retained after this step.

Labelling. The seven error types listed below were automatically induced. For incorporating certain types of errors in a sentence, specific sounds or characters need to be present.

1. **Vowel Stress Change.** If a vowel is not in stressed position,[9] change it to the respective vowel counterpart[10] (e.g. ‘кръгъл’ → ‘кръгал’).
2. **Assimilation.** If two neighbouring consonants differ in their voice quality,[11] change the former consonant so it follows the voice quality of the latter (e.g. ‘постановка’ → ‘постанофка’).
3. **Word-final Devoicing.** If there is a voiced consonant at the end of the word, change the consonant to its voiceless form (e.g. ‘масив’ → ‘масиф’).
4. **Double Consonant Loss.** If there is a double ‘т’ or double ‘н’, remove one of them (e.g. ‘пролетта’ → ‘преолета’).
5. **Consonant Clusters.** If a specified consonant cluster is present (e.g. ‘стн’, ‘здн’, ‘щт’), remove ‘т’ or ‘д’ (e.g. ‘местно’ → ‘месно’).
6. **Random Character.** Introduce a random character into a word (e.g. ‘момиче’ → ‘момгче’).
7. **Semantic Change.** If a character replacement, removal, addition or swap operation causes a word to result in a different word, which is spelled correctly, change it (e.g. ‘което’ (‘which’) → ‘котето’ (‘kitten’)).

It is worth noting that the resulting spelling correction dataset was not used for training any of the models presented in this work. Nevertheless, such a dataset is still necessary for quantifiable evaluation of any spelling correction model and it may prove useful to other members of the research community.

The data distribution presented in Figure 1 shows the frequencies of the different types of spelling errors within the dataset.

---

[9] A vowel in stressed position is pronounced longer and louder than an unstressed one.

[10] Bulgarian vowels are paired in terms of where they are articulated in the mouth, e.g. ‘а’ and ‘ъ’.

[11] Consonants in Bulgarian are separated into voiced and voiceless, with the majority of them forming pairs.

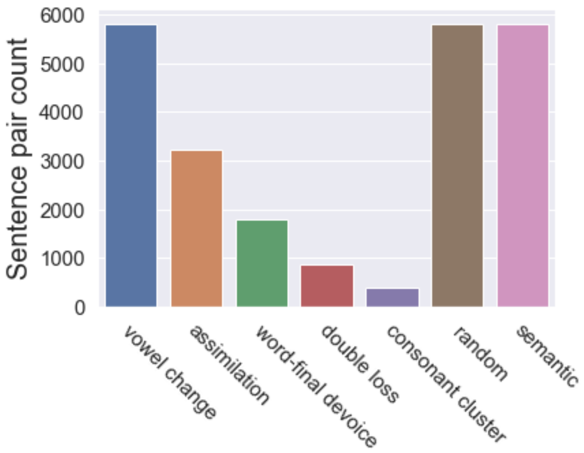


Figure 1: Error type distribution in the spelling error dataset.

3.2.3 Data set for Grammar Error Correction

Our grammar error dataset consists of 7588 error pairs. Similarly to the spelling error dataset, in each pair, the first sentence is the original correct sentence and the second one contains an induced error. Each erroneous sentence has only one induced grammatical error, which can be one of four error types.

Data Collection and Preprocessing. The source of the sentences is a combination of Wikipedia articles and Bulgarian data from the OSCAR project.[12] The same 28 Wikipedia articles in the spelling error dataset were used, in addition to 959,112 documents from the OSCAR dataset. The OSCAR documents were collected from open-source online materials, so the grammatical validity of the data may not be as good as Wikipedia. However, the Wikipedia articles are too similar in terms of writing style. Because of this, the errors induced from them are too similar; thus, including OSCAR diversified the dataset, allowing for a more balanced grammar error distribution.

Preprocessing steps that are similar to those applied on the spelling error dataset were used. Specifically, all documents were fed into a pipeline to perform sentence segmentation, tokenisation and POS tagging. Only sentences with three words or more and contained a token tagged as a verb were kept. Here, the POS tagging component of the CLaRK System (Simov et al., 2003) was used,[13] available via Mate Tools, a toolkit developed by University of Stuttgart's Institute for Natural Language Processing. [14]

	Part-of-Speech (POS) Tag Groups
1	{Ncmsf, Ncmsh}
2	{Pie-os-m, Pie-as-m}, {Pre-os-m, Pre-as-m}, {Prp–s-m, Prp–s-f, Prp–s-n, Prp–p}
3	{V-1p}
4	{Amsi, Afsi, Ansi, A-pi}, {Amsh, Afsd, Ansd, A-pd}, {Amsf, Afsd, Ansd, A-pd}

Table 1: POS tags used in the mappings for inducing different types of grammatical errors: (1) article misuse, (2) pronoun misuse, (3) incorrect verb suffix and (4) noun-adjective disagreement.

Labelling. The error types in the dataset fall under four types. The process of inducing grammatical errors is more sophisticated than in the case of the spelling error dataset, as the former required understanding of text that goes deeper than syntax.

Errors were induced by identifying a word with a source POS tag and then switching that word for a different inflected form with a target POS tag. The four error types are defined below.

1. **Article Misuse.** If there is a masculine noun with a definite article form, change it to its indefinite form, and vice-versa (e.g. 'синът' → 'сина').
2. **Pronoun Misuse.** If there is a pronoun, change its form:
    - with respect to the object/subject, similar to the use of 'I' and 'me' in English (e.g. 'който' → ' когото').
    - with respect to grammatical gender and/or count (e.g. 'чиито' → 'чийто').
3. **Incorrect Verb Suffix.** If there is a verb in the first person plural form that ends with 'м', append an 'е' (e.g. 'ядем.' → 'ядеме.').
4. **Noun-adjective Disagreement.** If there is a noun-adjective pair, introduce disagreement in terms of count and/or grammatical gender (e.g. 'красива' → 'красив').

The introduction of errors was implemented by defining mappings for source part-of-speech tags to target part-of-speech tags; these tags are provided (organised in one or multiple separate groups for each error) in Table 1. The mappings from source to target tag is generated by computing all possible combinations within each group; for instance, for

---

[12] https://oscar-project.org

[13] https://bultreebank.org/clark/bulgarian-nlp-pipeline-in-clark-system/

[14] https://www.ims.uni-stuttgart.de/en/research/resources/tools/matetools/

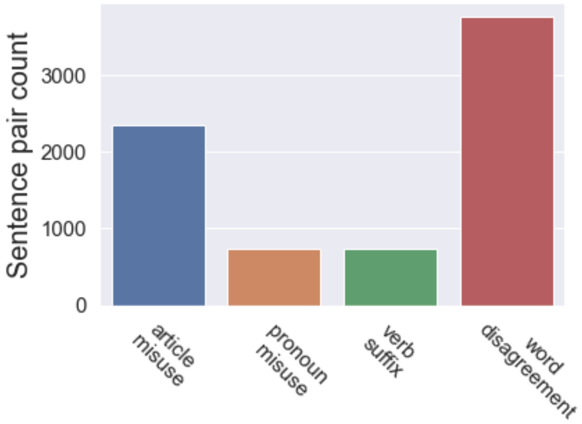


Figure 2: Error type distribution in the grammar error dataset.

article misuse, there is only one group with two POS tags. It follows then that there are only two possible mappings: {Ncmsf → Ncmsh, Ncmsh → Ncmsf}. The second error type contains three groups and overall defines 16 mappings (2 from the first, 2 from the second and 12 from the third). The tags follow the tagging scheme introduced by BulTreeBank (Simov et al., 2004). An exception to the aforementioned process is the third error type which regards incorrect verb suffix misuse. Here, there is only one group with only one relevant POS tag, which is used to identify verbs to append an incorrect suffix to, rather than build mappings from.

The distribution of grammatical error types in our dataset (Figure 2) is influenced by how common the relevant POS tags (corresponding to the error types) are.

4 Fine-tuning mT5 for GEC

4.1 Dataset Selection

For our GEC experiments, we decided to focus on only two of the four error classes we defined, i.e., article and pronoun mistakes, as these are considered to be the most prevalent errors in Bulgarian writing. Filtering the examples based on these error types left us with 3297 pairs. Out of these, we retained only the pairs where neither of the sentences exceeded a 300-character limit, as we consider any sentences longer than that to be anomalous. In the end, 3090 pairs remained. This dataset was utilised for model training and evaluation, whereby subsets with 72%, 18% and 10% of the data were used for training, validation and testing, respectively.

4.2 Model Training

Whereas the original T5 (Raffel et al., 2020) model works exclusively for English, the mT5 model supports multilingual text. Our proposed approach is based on fine-tuning the mT5 model, which has previously demonstrated encouraging performance for GEC in other highly inflected languages (Lytvyn et al., 2023). Specifically, we employed the trained mT5 model available from Huggingface.[15]

During the training stage, an mT5 model takes two sequences, i.e., the source and the target, and learns to transform the first into the second. In our case this would have the source sequence as a sentence with an error and the target sequence would be the same sentence, but corrected. An example is given below.

- Source (erroneous): ‘Емануела седна на столът.’
- Target (correct): ‘Емануела седна на стола.’

The translation for both is ‘Emanuela sat on the chair.’ However, in the source sequence, the word ‘chair’ (‘столът’) is used in its definite form, instead of indefinite (‘стола’). This constitutes a grammatical mistake, as only the subject of the sentence should be used in its definite form.

In order to determine the most optimal values of training hyperpameters, we conducted grid search, whereby the search space was defined based on the hyperparameter values below.

- weight decay rate: {0.1, 0.01, 0.001}
- learning rate: {0.00002, 0.0002, 0.002}
- training batch size: {4, 8}

All 18 hyperparameter combinations were used in fine-tuning the mT5 model for 16 epochs. The process was repeated three times to allow us to take the average over the results, ensuring stable performance.

As can be observed in the visualisation in Figure 3 which presents the validation loss according to the hyperparameter values, a learning rate of 0.002 seems too high, with lower rates yielding better performance. Given this, we performed an experiment to compare the other two learning rates, 0.0002 and 0.00002 (see Figures 6 and 7 in the Appendix). Upon using these two learning rates, it became evident that the former is a better choice. Following this, the final hyperparameter combination that we chose is: learning rate = 0.0002, weight decay = 0.01, batch size = 8. We also decided to fix the

---

[15] `https://huggingface.co/google/mt5-base`

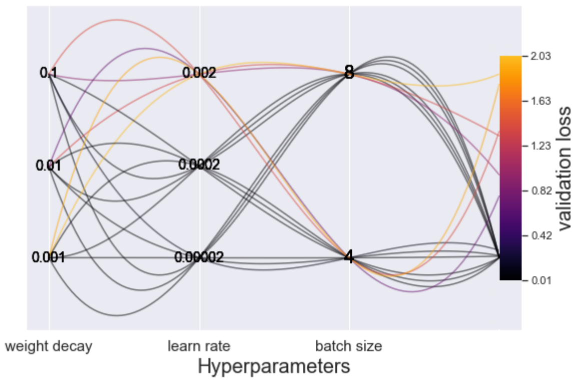


Figure 3: Validation loss for various hyperparameter values: weight decay, learning rate and batch size.

number of epochs to 4, as our experiments showed that this leads to the lowest validation loss.

5 Evaluation

5.1 Dictionary

Our dictionary contains both lemmas and inflected forms, unlike the official one released by The Institute for Bulgarian Language which only contains lemmas. As a means for evaluating its coverage, we randomly selected 20,000 entries from our dictionary. These were then checked against the official dictionary and it was found that only 1292 were present. Within the 93.51% of the missing words, most are inflected versions of base word forms. This only goes to show how existing dictionary resources do not exhibit sufficient coverage for spell-checking tasks.

5.2 GEC Model

Evaluation of the model is performed by comparing its performance on the GEC task with two contemporary large language models (LLMs) that can handle Bulgarian: gpt3.5-turbo[16] and BgGPT.[17] Despite its name which implies that it is based on GPT (Brown et al., 2020), BgGPT is in fact a fine-tuned Mistral model (Jiang et al., 2023). Both of our chosen models for comparison are decoder-only transformers, and rely solely on autoregressive generation. Ideally, evaluation should be performed using other encoder-decoder models; however, no suitable alternatives that can handle Bulgarian were found.

Our evaluation involves both quantitative and qualitative comparison, utilising standard metrics and a survey among human participants, respectively.

5.2.1 Quantitative Evaluation

The two LLMs were evaluated based on the commonly used strategy of zero-shot prompting (Rosa et al., 2021), whereby no training examples are provided to the models prior to evaluation. Both models were prompted with each input example from the test set (309 sentence pairs) three times, averaging metric results to account for randomness. Additionally, as prompt engineering has been shown to greatly impact LLM responses (Marvin et al., 2023), two different prompts were utilised.

1. *Correct the mistake: [erroneous sentence]*
2. *Look at the following sentence and rewrite it, fixing any mistakes if there are any: [erroneous sentence]*

The performance of the models, including our fine-tuned mT5 model, is presented in Table 2 in terms of precision, recall and F0.5-score (i.e., F_β score, where $\beta = 0.5$). GEC models are typically evaluated with this F-score setting since the CoNLL-2014 shared task on GEC (Ng et al., 2014), because a lower β score places a higher emphasis on precision, i.e., scoring higher for ensuring predicted tokens are correct, rather than correcting all mistakes.

In this scenario, a true positive (TP) is an erroneous token that has been replaced by its corrected version with respect to the gold reference data. Meanwhile, a false positive (FP) is a correct token being wrongly replaced and a false negative (FN) is an erroneous token that remains unchanged. If a token is erroneous but has been replaced with a token that is not the one specified by the gold standard, it counts both as an FP and an FN.

Our proposed fine-tuned mT5 model outperforms all variants of the contemporary models. Both gpt3.5-turbo and BgGPT scored high on recall, i.e., they corrected a majority of the errors. However, a low precision score implies they tend to over-correct. Their ‘corrections’ oftentimes do not introduce new errors; they simply reword the source sentence. Nevertheless, they were prompted to only correct errors and rewording runs the risk of changing the sentence semantically.

The proposed fine-tuned mT5 model is not only able to obtain higher recall than both models, but

---

[16] https://platform.openai.com/docs/models/gpt-3-5-turbo

[17] https://huggingface.co/INSAIT-Institute/BgGPT-7B-Instruct-v0.1

	Precision	**Recall**	**F0.5-score**
gpt3.5-turbo (prompt #1)	37.51 (± 5.40)	60.52 (± 5.45)	39.34 (± 5.45)
BgGPT (prompt #1)	33.07 (± 5.25)	59.87 (± 5.47)	35.03 (± 5.32)
gpt3.5-turbo (prompt #2)	38.62 (± 5.43)	66.02 (± 5.28)	40.74 (± 5.48)
BgGPT (prompt #2)	30.18 (± 5.19)	62.33 (± 5.40)	32.33 (± 5.26)
Fine-tuned mT5 (Ours)	**68.12** (± 5.20)	**68.61** (± 5.17)	**68.18** (± 5.19)

Table 2: Comparison of models for GEC, including 95% confidence intervals.

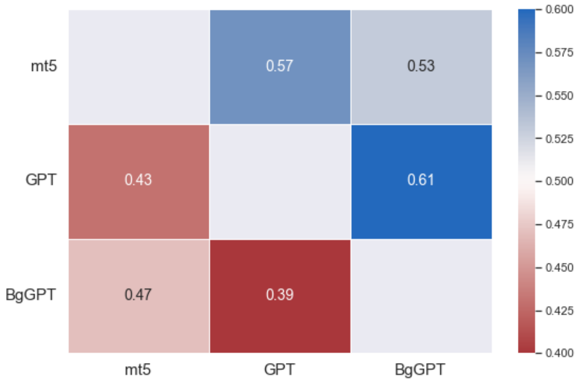


Figure 4: Proportion of pairwise model preference counts for fine-tuned mT5, gpt3.5-turbo and BgGPT. Rows indicate the winners (preferred model) and columns correspond to their respective opponents.

also outperforms them across all metrics, especially in terms of F0.5-score. This ensures that the model actively attempts to exclusively correct errors.

5.2.2 Qualitative Evaluation

To assess the correctness, readability and understandability of our proposed solution, we have conducted a survey to compare the performance of the three GEC models from the previous section: BgGPT, gpt3.5-turbo and our fine-tuned mT5 model.

Design. Examples in the test set were used to prompt the proposed solution, as well as BgGPT and gpt3.5-turbo, based on prompt #1, resulting in 309 sentence triplets. The survey included only triplets where all three model predictions are different from one another. In the survey, 13 questions were presented: the first 8 were related to article misuse and the last 5 focussed on pronoun misuse. We refer the reader to Figures 8 and 9 in the Appendix for examples of questions presented to participants as part of our survey.

Results. Overall, 67 Bulgarian native speakers completed the survey. They were recruited by contacting Bulgarian social media groups and AI communities in Bulgaria. Each response served as an indication of pairwise preferences, resulting in 2613 comparisons, provided in Table 3 in the Appendix.

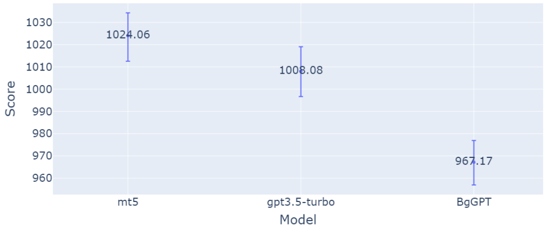


Figure 5: Bradley-Terry scores from survey rankings for our fine-tuned mT5 model, gpt3.5-turbo and BgGPT with 95% confidence intervals.

Our fine-tuned mT5 model obtained the highest preference count, with a total of 963. Its preference count proportions, visualised in Figure 4 (57% and 53% vs GPT and BgGPT, respectively) are higher than those of the respective alternative models.

Inspired by the ChatBot Arena[18] (Zheng et al., 2024), we built a Bradley-Terry model to assign a score with confidence intervals to each GEC model based on the responses. As shown in Figure 5, our fine-tuned mT5 model was able to perform better than the contemporary LLMs with a statistically significant difference.

6 Conclusion

This paper presents a state-of-the-art solution for Bulgarian GEC based on the encoder-decoder transformer model mT5, which surpasses LLMs like gpt3.5-turbo and BgGPT. Additionally, we present a contribution in the form of datasets supporting Bulgarian language literacy, including a machine-readable dictionary and two datasets with erroneous-corrected sentence pairs: one for spelling and the other for grammar mistakes.

Future work could investigate additional specialised initiatives regarding the collection of natural learner data from Bulgarian learners. Additionally, language literacy entails punctuation; there is active NLP research in restoring and correcting punctuation in texts (Gravano et al., 2009; Tekir et al., 2023), which can be explored in Bulgarian.

---

[18] `https://chat.lmsys.org`

Limitations

Whilst the dictionary collection was evaluated for its coverage, the error correction datasets were generated automatically assuming that the source text is correct.

Our proposed GEC model was trained to identify and correct Bulgarian grammar errors that are based on article and pronoun misuse. Thus, in its current version, it is unlikely to perform well on other types of errors. In addition, the model was not trained to be correct in terms of facts pertaining to people or events, and therefore using the model to generate such content is out-of-scope.

Ethics Statement

The presented datasets and models utilise open-source and publicly available resources (e.g., Wikipedia, OSCAR) that do not contain the names, contact information, addresses, birth dates or other information that can be considered private and/or sensitive.

The survey that we conducted to qualitatively evaluate GEC models did not require users to provide any personal information and no such data was collected for this project.

References

N. Abdul-Rahaman, E. Terentev, and V.E. Arkorful. 2023. The Tertiary Experience: Of Social Integration, Retention and Persistence–A Review. *Public Organization Review, 23(1)*, pages 133–147.

Melania Berbatova and Filip Ivanov. 2023. An Improved Bulgarian Natural Language Processing Pipeline. *Annual of Sofia University St. Kliment Ohridski. Faculty of Mathematics and Informatics*, 110:37–50.

Nadezhda Borisova, Grigor Iliev, and Elena Karashtranova. 2014. On Detecting Noun-Adjective Agreement Errors in Bulgarian Language Using GATE. *Preprint*, arXiv:1411.0588.

Svetla Boytcheva, Boris Velichkov, Gerasim Velchev, and Ivan Koychev. 2020. Automatic Generation of Annotated Corpora of Diagnoses with ICD-10 codes based on Open Data and Linked Open Data. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 163–167.

Chris Brockett, Bill Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, Australia*, pages 249—-256.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33:1877–1901.

Dina C Castro, Mariela M Páez, David K Dickinson, and Ellen Frede. 2011. Promoting Language and Literacy in Young Dual Language Learners: Research, Practice, and Policy. *Child Development*, 5(1):15–21.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5755–5762.

Denitsa Dimitrova. 2021. Bulgarian Speech Corpora: A Review. In *International CLaDA-BG Conference 2021*, pages 3–58.

Natalia Dolgova and Charles Mueller. 2019. How useful are corpus tools for error correction? insights from learner data. *Journal of English for Academic Purposes*, 39:97–108.

J. Geertzen, T. Alexopoulou, and A. Korhonen. 2013. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCamDat). *31st Second Language Research Forum. Somerville, MA: Cascadilla Proceedings Project*, pages 240–254.

Stefan Gerdjikov, Stoyan Mihov, and Vladislav Nenchev. 2013. Extraction of Spelling Variations from Language Structure for Noisy Text Correction. In *2013 12th International Conference on Document Analysis and Recognition*, pages 324–328.

S. Granger, E. Dagneaux, F. Meunier, and M. eds. Paquot. 2009. *International corpus of learner English (Vol. 2)*. Presses Universitaires de Louvain.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4741–4744.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-Augmented Grammatical Error Correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 252–263.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic Error Detection in the Japanese Learners' English Spoken Data. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Svetla Koeva, Nikola Obreshkov, and Martin Yalamov. 2020. Natural Language Processing Pipeline to Annotate Bulgarian Legislative Documents. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6988–6994.

Vladislav Kubon and Martin Plátek. 1994. A Grammar Based Approach to a Grammar Checking of Free Word Order Languages. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, pages 906–910.

John Lee and Stephanie Seneff. 2008. Correcting Misuse of Verb Forms. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL-08: HLT*, pages 174–182.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora Generation for Grammatical Error Correction. In *Proceedings of the 2019 Conference of the North*, pages 3291—-3301.

Vasyl Lytvyn, Petro Pukach, Victoria Vysotska, Myroslava Vovk, and Nataliia Kholodna. 2023. Identification and Correction of Grammatical Errors in Ukrainian Texts Based on Machine Learning Technology. *Mathematics*, 11(4):904–923.

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt Engineering in Large Language Models. In *International Conference on Data Intelligence and Cognitive Informatics*, pages 387–402.

Jürgen M. Meisel. 2020. Zisa dataset.

National Statistical Institute. 2023. Статистически справочник 'Преброяване 2021' *[Statistical reference book 'Census 2021']*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the eighteenth conference on computational natural language learning: shared task*, pages 1–14.

OECD. 2023. *PISA 2022 Results (Volume I)*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research*, 21(140):1–67.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51.

Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. To tune or not to tune? zero-shot models for legal case entailment. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 295–300.

Alla Rozovskaya and Dan Roth. 2019. Grammar Error Correction in Morphologically Rich Languages: The Case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

K. Simov, P. Osenova, and M. Slavcheva. 2004. BTB-TR03: BulTree-Bank Morphosyntactic Tagset. BTB-TS version 2.0. Technical report, Bulgarian Academy of Sciences.

Kiril Simov, Gergana Popova, and Petya Osenova. 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). *A rainbow of corpora: Corpus linguistics and the languages of the world*, pages 135–142.

Kiril Simov, Alexander Simov, Milen Kouylekov, Krasimira Ivanova, Ilko Grigorov, and Hristo Ganev. 2003. Development of corpora within the CLaRK system: The BulTreeBank project experience. In *Demonstrations*, pages 243–246.

Aiman Solyman, Zhenyu Wang, and Qian Tao. 2019. Proposed Model for Arabic Grammar Error Correction Based on Convolutional Neural Network. In *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–6.

Aiman Solyman, Marco Zappatore, Wang Zhenyu, Zeinab Mahmoud, Ali Alfatemi, Ashraf Osman Ibrahim, and Lubna Abdelkareim Gabralla. 2023. Optimizing the impact of data augmentation for low-resource grammatical error correction. *Journal of King Saud University-Computer and Information Sciences*, 35(6):101572.

I.M. Suarta, I.K. Suwintana, I.F.P. Sudhana, and N.K.D. Hariyanti. 2017. Employability Skills Required by the 21st Century Workplace: A Literature Review of Labor Market Demand. *International Conference on Technology and Vocational Teachers (ICTVT 2017)*, pages 337–342.

Selma Tekir, Aybüke Güzel, Samet Tenekeci, and Bekir Haman. 2023. Quote Detection: A New Task and

Dataset for NLP. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 21–27.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A Comprehensive Survey of Grammar Error Correction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–51.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data. *Preprint*, arXiv:1903.00138.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-Judge with MT-Bench and Chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Appendix

	mT5	GPT	BgGPT
mT5	-	500	463
GPT	371	-	529
BgGPT	408	342	-

Table 3: Pairwise preference counts across the GEC models. Rows indicate the winners (preferred model) and columns correspond to their respective opponents.

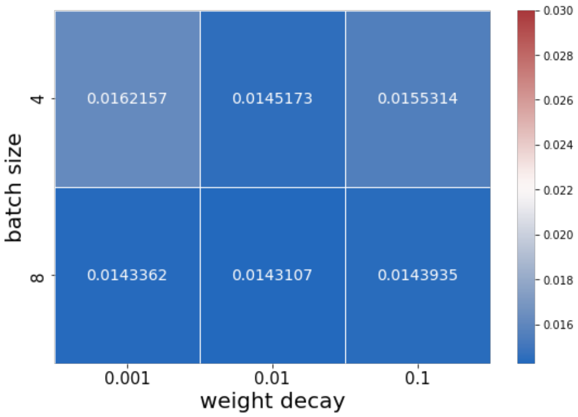


Figure 6: Validation loss obtained by our fine-tuned mT5 model, using a learning rate of 0.0002.

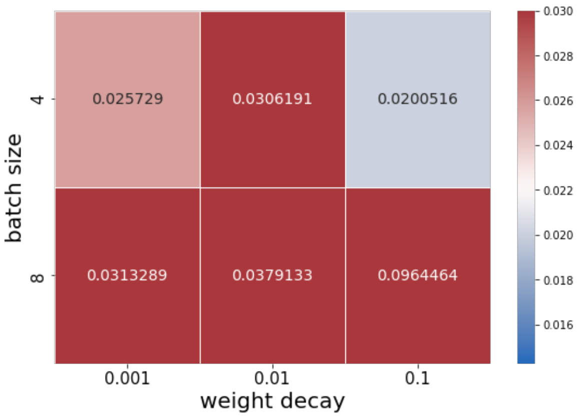
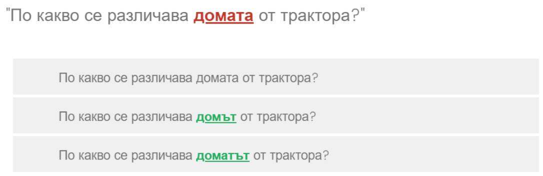


Figure 7: Validation loss obtained by our fine-tuned mT5 model, using a learning rate of 0.00002.



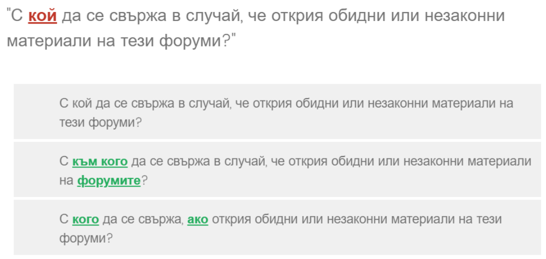
"По какво се различава **домата** от трактора?"

По какво се различава домата от трактора?

По какво се различава **домът** от трактора?

По какво се различава **доматът** от трактора?

Figure 8: Survey question asking a participant to rank a correction for an article misuse error.



"С **кой** да се свържа в случай, че открия обидни или незаконни материали на тези форуми?"

С кой да се свържа в случай, че открия обидни или незаконни материали на тези форуми?

С **към кого** да се свържа в случай, че открия обидни или незаконни материали на **форумите**?

С **кого** да се свържа, **ако** открия обидни или незаконни материали на тези форуми?

Figure 9: Survey question asking a participant to rank a correction for a pronoun misuse error.

On Barriers to Archival Audio Processing

Peter Sullivan
University of British Columbia, Canada
`prsull@student.ubc.ca`

Muhammad Abdul-Mageed
University of British Columbia, Canada
`muhammad.mageed@ubc.ca`

Abstract

In this study, we leverage a unique UNESCO collection of mid-20th century radio recordings to probe the robustness of modern off-the-shelf language identification (LID) and speaker recognition (SR) methods, especially with respect to the impact of multilingual speakers and cross-age recordings. Our findings suggest that LID systems, such as Whisper, are increasingly adept at handling second-language and accented speech. However, speaker embeddings remain a fragile component of speech processing pipelines that is prone to biases related to the channel, age, and language. Issues which will need to be overcome should archives aim to employ SR methods for speaker indexing.

1 Introduction

Multinational organizations such as the United Nations (UN); the International Federation of Red Cross and Red Crescent Societies (IFRC); and the United Nations Educational, Scientific, and Cultural Organization (UNESCO) maintain audio archives that are of profound cultural and historical value. However, incomplete descriptive metadata often hinder their access by the public (Zervanou et al., 2011).

Audio archives present a complex terrain for contemporary speech processing technologies, owing to the varied domains these recordings encapsulate. The long running MALACH project's effort to tackle emotional, disfluent, and accented speech (Picheny et al., 2019; Psutka et al., 2002) gives of sense of this complexity. In this study, our attention is directed towards extensive multilingual repositories, which pose challenges for speaker recognition (SR) and language identification (LID) technologies due their long chronological span and inclusion of second-language (L2) speech.

A primary objective in enhancing accessibility to these recordings involves identifying the speakers within a specific recording. While documentation of speakers is *sometimes* available, it is only at the document level, making this task closer to speaker indexing (Sturim et al., 2001) than standard SR. Moreover, concerns regarding the robustness of speaker embeddings to voice modifications associated with aging, as well as the accuracy of language-specific predictions (Hutiri and Ding, 2022), significantly challenge the straightforward utilization of off-the-shelf SR technologies within the ambit of these long-running, multilingual archives.

In this exploratory investigation using a selection of radio audio archives from UNESCO, we explore the impact of these factors on robustness of zero-shot application of off-the-shelf tools, to identify paths towards speaker indexing in age- and language-variable environments. Our dataset involves 171 hours of archival data covering the period of 1952-1980, involving 20 languages (See Table 1). Our work offers the following contributions:

1. We characterize a relatively neglected sphere within speech processing scholarship: multilingual audio archives.
2. We carry out a cross-age analysis to investigate robustness of the speaker embeddings.
3. We analyze the robustness of speaker embeddings in multilingual speech scenarios, uncovering unique insights that are otherwise hidden without access to datasets such as ours.
4. We investigate the utility of a number of off-the-shelf language identification tools for accented LID.

The rest of the paper is organized as follows: Section 2 offers a brief overview of speech processing as it relates to challenges present in archival audio. Section 3 is a description of our datasets.

Section 4 details our data processing methods. Section 5 gives an overview of the LID and SR models used. Section 6 is a breakdown of our different experiments. In Section 7, we provide our experimental results and discuss these in Section 8. We discuss limitations of our work in Section 9 and provide ethical implications in Section 10. We conclude in Section 11.

2 Background

2.1 Language ID

Recent work in LID has moved from the discrete segment representations popular in earlier i-vector (Dehak et al., 2011) and x-vector (Snyder et al., 2018) works, to the convenience of end-to-end deep neural models either based on ResNets (Cai et al., 2018) or Transformers (Babu et al., 2021; Radford et al., 2023; Pratap et al., 2023). Part of this has been enabled by the creation of LID datasets such as VoxLingua107 (Valk and Alumäe, 2021) and FLEURS (Conneau et al., 2023), allowing for direct training of large end-to-end models. Meanwhile, another aspect of this evolution has been the inclusion of LID into models designed as 'jack-of-all-trades' tools such as Whisper (Radford et al., 2023) and Massively Multilingual Speech (MMS) (Pratap et al., 2023), which are built to support automatic speech recognition (ASR) and speech translation in addition to LID.

Recent work in LID has raised attention to less well developed areas of exploration including L2 LID (Kukk and Alumäe, 2022), LID for multilingual users (Titus et al., 2020), and domain generalization of LID (Sullivan et al., 2023). For archival audio, these three aspects become even more relevant, as the long chronological nature of the archives and diverse population of multilingual speakers demand the use of highly robust LID systems. At the same time, the operational context (including lack of funding and know-how) means that off-the-shelf tools are the only viable option, neglecting transfer learning or domain adaptation approaches, regardless of their efficacy.

2.2 Speaker embedding

Modern SR makes use of speaker embeddings (i.e. x-vector, r-vector etc.) often extracted from ResNets-based extractor networks (Desplanques et al., 2020; Wang et al., 2023) and trained on VoxCeleb (Nagrani et al., 2017). Despite the progress made in the field, a recent study of bias in SR (Hutiri and Ding, 2022) indicate a number of issues, including demographic biases impacting age, gender, language, and nationality. To our knowledge little work has been performed to investigate the impact of language (including multilinguality) on SR effectiveness. However, there is a body of work that has looked at the impact of cross-age scenarios on SR (Singh et al., 2023; Qin et al., 2022). With Qin et al. (2022) employing adversarial learning to learn age-invariant embeddings, and Singh et al. (2023) estimating the impact of age on cross-age verification. To our knowledge no work has considered investigation of cross-age scenarios in multilingual corpora.

Work in speaker indexing, has been largely neglected since anchor model based approaches (Sturim et al., 2001), but we hope that this study will pave the way to future work in speaker indexing by identifying existing obstacles with off-the-shelf tools.

Table 1: Dataset Overview. n is the number of recordings. For LID, we use only the first 30 seconds of spoken audio from each recording.

Split	n	Hours	Languages
LID	484	4	20
Cross-age	692	104	17
Cross-lingual	463	67.5	20

3 Datasets

VoxPopuli - L2 English An existing source of L2 audio, in a similar domain to the one under consideration is the VoxPopuli collection of EuroParl recordings (Wang et al., 2021). In particular we are interested in the 29 hour subset of L2 English speech coming from 15 different accent backgrounds.

Radio Archives From our partnership with UNESCO, we utilize a mid-20th century radio recording archives collection that is currently undergoing metadata enrichment. While the dataset is currently not published, the metadata used and links to the recordings will be made available. The recordings in this collection consist of a variety of material including speeches, interviews, news briefs, and educational programs. The audio was recorded between the 1950s and early 1980s, and digitized in the late 2010s. See Table 1 for more detailed statistics.

4 Data Processing

While the VoxPopuli dataset is used with existing segmentation for our L2 English experiments, we build two datasets from the radio archives by filtering the known metadata to meet certain criteria. As much of the radio archives contain incomplete metadata, we restrict our selection to recordings identified as having a single known speaker on the recording,as well as a single languages spoken.

From this selection, we create two datasets: Our *cross-age* dataset is filtered by selecting speakers with multiple recordings in the same language occurring in different calendar years. The second *cross-lingual* dataset is filtered by selecting speakers with multiple recordings in different languages.

For both datasets, we further filter by diarizing the recordings using Pyannote's (Bredin et al., 2020) speaker diarization pipeline (version 3.1), and selecting recordings where there is a single speaker accounting for more than 75% of the duration. We utilize 16khz single channel copies of the recordings for the study.

5 Models

Whisper (Radford et al., 2023) is a multilingual speech processing model that allows for ASR, speech translation, and LID. We use all three versions of the large (1.5B parameter) model.

MMS (Pratap et al., 2023) is similarly a multilingual speech processing model allowing for ASR, speech translation, LID, and additionally text-to-speech. The model has been primarily trained on the Bible and other religious audio, with a focus on scaling the number of covered languages. We use the (1B parameter) 126 language version.

WeSpeaker ResNet34-LM (Wang et al., 2023) is a SR model trained on VoxCeleb (Nagrani et al., 2017).

6 Experiments

6.1 L2 LID

For our L2 LID experiments, we compare two well known off-the-shelf LID models: Whisper Large V(1-3) (Radford et al., 2023) and MMS L126 (Pratap et al., 2023). For our VoxPopuli dataset we simply report predictions on the given segments. However, for our cross-lingual radio dataset, we follow the suggested LID procedure of the Whisper model and select the first 30 seconds of audio to perform the prediction.

6.2 Speaker Embedding Robustness

To understand how robust speaker embeddings are to cross-age and cross-lingual effects, we use a pretrained ResNet model, Wespeaker ResNet34-LM (Wang et al., 2023), to extract segment representations. We take the average embedding for the majority speaker in each recording. For the *cross-age experiments*, we calculate the cosine similarity between representations and aggregate by difference in calendar years between recordings. We rely on calendar year as an approximation as we do not have recording dates for some recordings. For the *cross-lingual experiments*, we simply compare the cosine similarity between same language segments to their different language similarity scores.

7 Results

For L2 English performance (see Table 2 in Appendix A), The updated Whisper V3 model substantially improves the recognition of accented English audio attaining an accuracy of 94% compared to both Whisper V1,V2 and MMS L126.

For our mixed L1 and L2 multilingual archival audio (see Table 3 in Appendix A), we see similar results as to the VoxPopuli results. Notably, MMS performs better on the mixture of accented and unaccented speech (accuracy at 71.90%). Still, the Whisper V3 model performs best on this set of audio: accuracies of 88.01% for V1, 87.60 for V2, and 91.32% for V3.

Looking at the robustness of the speaker embeddings, we notice a substantial drop in the similarity scores in both the cross-age setting (see Fig. 1) and cross-lingual setting (see Fig. 2). For the cross-age comparison, median cosine similarity scores continue to drop until stabilizing after a gap of 10 years. While cross-lingual performance of representations drops substantially as well, of note is the very large increase in standard deviation between the two settings. This latter trend is potentially problematic for treating all cross-lingual scenarios the same, and may be indicative that fluency as well as language similarity between compared languages may be factors.

8 Discussion

The performance of the Whisper V3 model appears to indicate it as a strong candidate for archives in processing multilingual audio files. Notably, while all of the Whisper models under examination have roughly the same number of parameters (V3 is

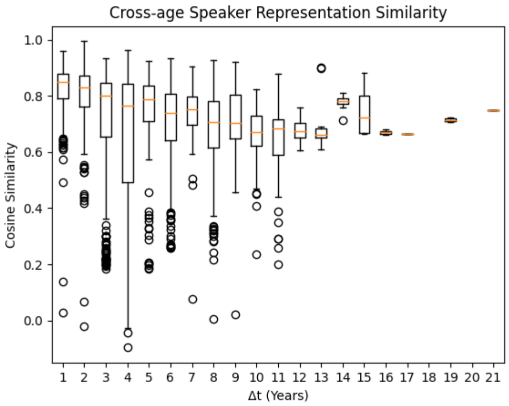


Figure 1: Longitudinal comparison of speaker embedding. Cosine similarity scores are plotted with median and quartiles marked. Outliers are noted as circles. Δt is the absolute difference in calendar years between each pair of recordings. Data past 15 years becomes quite sparse, as few speakers fit our filtering criteria.

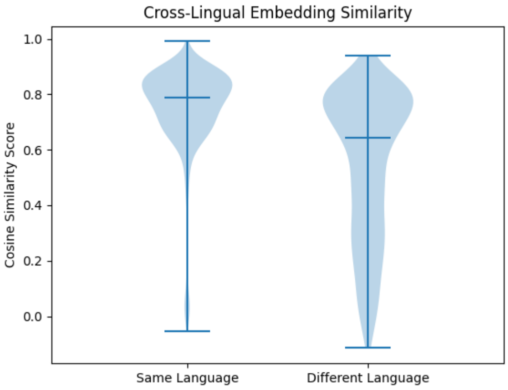


Figure 2: Cross-lingual comparison of speaker embedding cosine similarity scores. To better show the distribution, we present the results as violin plots, noting that the lower end of the Same Language plot may be representative of different speakers who were not filtered out of our automated approach. Distribution statistics: *Same language* Mean: 0.71, Median: 0.76, Std: 0.19; *Different language* Mean: 0.53, Median: 0.60, Std: 0.26

slightly larger due to increasing the input dimensions), the larger amount of training data used for V3 appear to substantially help it in identifying accented English. The brittle nature of the MMS model with regard to L2 speech is quite surprising, and demonstrates the importance of having diverse and challenging audio benchmarks for LID. While the longitudinal embedding comparison demonstrates the challenges of applying SR models across channel and age. The appearance of a flattening at around 10 years of difference indicates that it may be possible to account for this cross-age drift.

9 Limitations

Working with real world datasets presents substantial challenges and limitations. While the archival partner aims to put the radio recordings online sometime in the future, these are not yet available publicly. Similarly, working with aggregated data has limitations. For instance the small cluster of low similarity in the same language embedding comparison is likely indicative of misidentified speakers who slipped through the filtering process.

10 Ethics

Speaker identification tools have the potential for significant privacy violation, especially if applied as part of a surveillance system at scale. As seen from the study, the large cross-channel and cross language variability in speaker embeddings present significant risks for use in larger scales, where risk associated with misidentification may also be quite high.

We believe our investigation of these tools for the purpose of improving archival accessibility and discovery are consistent with ethical practice, and our application to a limited closed set of public figures on public audio recordings further limits this potential for abuse.

11 Conclusion

We present an analysis into the robustness of SR and LID tools on cross-age, and cross-lingual audio on a unique archival radio dataset. Our work indicates the viability of existing LID tools, such as Whisper V3 (Radford et al., 2023), at handling the accented and multilingual speech common in recordings from multinational organizations. We also find that cross-age and cross-lingual application of SR introduce large drops in the cosine similarity scores, with the cross-age similarity dropping steadily over a period of a decade. The cross-lingual scores further introducing large increase in the standard deviation, potentially indicative of other factors such as language fluency or language similarity impacting the result. Additionally, this work demonstrates the value of archival audio in examining current speech processing approaches. The variety of such recordings offer a strong platform to study bias and domain adaptation.

References

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. XLS-R: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. 2020. Pyannote. audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128. IEEE.

Weicheng Cai, Zexin Cai, Xiang Zhang, Xiaoqi Wang, and Ming Li. 2018. A novel learnable dictionary encoding layer for end-to-end language identification. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5189–5193. IEEE.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Najim Dehak, Pedro A. Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak. 2011. Language recognition via i-vectors and dimensionality reduction. In *Proc. Interspeech 2011*, pages 857–860.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Proc. Interspeech 2020*, pages 3830–3834.

Wiebke Toussaint Hutiri and Aaron Yi Ding. 2022. Bias in automated speaker recognition. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 230–247.

Kunnar Kukk and Tanel Alumäe. 2022. Improving Language Identification of Accented Speech. In *Proc. Interspeech 2022*, pages 1288–1292.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. VoxCeleb: A large-scale speaker identification dataset. In *Proc. Interspeech 2017*, pages 2616–2620.

Michael Picheny, Zoltán Tüske, Brian Kingsbury, Kartik Audhkhasi, Xiaodong Cui, and George Saon. 2019. Challenging the Boundaries of Speech Recognition: The MALACH Corpus. In *Proc. Interspeech 2019*, pages 326–330.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Josef Psutka, Pavel Ircing, Josef V Psutka, Vlasta Radová, William J Byrne, Jan Hajič, Samuel Gustman, and Bhuvana Ramabhadran. 2002. Automatic transcription of Czech language oral history in the MALACH project: Resources and initial experiments. In *Text, Speech and Dialogue: 5th International Conference, TSD 2002 Brno, Czech Republic, September 9–12, 2002 Proceedings 5*, pages 253–260. Springer.

Xiaoyi Qin, Na Li, Weng Chao, Dan Su, and Ming Li. 2022. Cross-Age Speaker Verification: Learning Age-Invariant Speaker Embeddings. In *Proc. Interspeech 2022*, pages 1436–1440.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Vishwanath Pratap Singh, Md Sahidullah, and Tomi Kinnunen. 2023. Speaker Verification Across Ages: Investigating Deep Speaker Embedding Sensitivity to Age Mismatch in Enrollment and Test Speech. In *Proc. INTERSPEECH 2023*, pages 1948–1952.

David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. Spoken Language Recognition using X-vectors. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 105–111.

Douglas E Sturim, Douglas A Reynolds, Elliot Singer, and Joseph P Campbell. 2001. Speaker indexing in large audio databases using anchor models. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (cat. no. 01ch37221)*, volume 1, pages 429–432. IEEE.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the Robustness of Arabic Speech Dialect Identification. In *Proc. INTERSPEECH 2023*, pages 5326–5330.

Andrew Titus, Jan Silovsky, Nanxin Chen, Roger Hsiao, Mary Young, and Arnab Ghoshal. 2020. Improving language identification for multilingual speakers. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8284–8288. IEEE.

Jörgen Valk and Tanel Alumäe. 2021. VoxLingua107: a dataset for spoken language recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 652–658. IEEE.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*.

Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. 2023. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Kalliopi Zervanou, Ioannis Korkontzelos, Antal Van Den Bosch, and Sophia Ananiadou. 2011. Enrichment and structuring of archival description metadata. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 44–53.

Appendix A LID Experiments

Table 2: L2 English LID Performance of Whisper (Radford et al., 2023) and MMS (Pratap et al., 2023)

Model	Accuracy
Whisper-Large v1	72.65%
Whisper-Large v2	72.65%
Whisper-Large v3	94.52%
MMS L126	11.10%

Table 3: Multilingual Archival Audio LID Performance of Whisper (Radford et al., 2023) and MMS (Pratap et al., 2023)

Model	Accuracy
Whisper-Large v1	88.01%
Whisper-Large v2	87.60%
Whisper-Large v3	91.32%
MMS L126	71.90%

Conversational Exploratory Search of Scholarly Publications Using Knowledge Graphs

Phillip Schneider and **Florian Matthes**
Technical University of Munich, Department of Computer Science, Germany
`{phillip.schneider, matthes}@tum.de`

Abstract

Traditional search methods primarily depend on string matches, while semantic search targets concept-based matches by recognizing underlying intents and contextual meanings of search terms. Semantic search is particularly beneficial for discovering scholarly publications where differences in vocabulary between users' search terms and document content are common, often yielding irrelevant search results. Many scholarly search engines have adopted knowledge graphs to represent semantic relations between authors, publications, and research concepts. However, users may face challenges when navigating these graphical search interfaces due to the complexity and volume of data, which impedes their ability to discover publications effectively. To address this problem, we developed a conversational search system for exploring scholarly publications using a knowledge graph. We outline the methodical approach for designing and implementing the proposed system, detailing its architecture and functional components. To assess the system's effectiveness, we employed various performance metrics and conducted a human evaluation with 40 participants, demonstrating how the conversational interface compares against a graphical interface with traditional text search. The findings from our evaluation provide practical insights for advancing the design of conversational search systems.

1 Introduction

Digital publication platforms have greatly expanded the accessibility of scholarly articles, offering an extensive range of publications that can be efficiently discovered through integrated search engines. These digital platforms provide researchers with access to millions of scholarly documents, encompassing conference papers, journal articles, workshop proceedings, and book chapters. The number of scholarly documents is growing exponentially, with estimates suggesting that it doubles approximately every 15 years (Bornmann et al., 2021). As the body of literature grows, traditional keyword-based search methods are becoming less effective at filtering and ranking relevant documents. These lexical methods rely heavily on well-formulated queries and otherwise yield irrelevant results. Researchers are often hindered by the so-called *vocabulary mismatch problem*, which manifests as differences between search terms and the terminology in the documents (Furnas et al., 1987). This issue is especially pronounced in open-ended and exploratory search scenarios, where users navigate unfamiliar information spaces. In such scenarios, users' incomplete knowledge of certain topics prevents them from formulating queries to access the information they need (Schneider et al., 2023a).

Reacting to the challenges posed by the high volume of scientific output, digital publication platforms have begun to make their search functionalities more intelligent by employing semantic search methods using natural language processing (NLP). These methods enable search engines to understand the context and intent behind user queries. Moving beyond exact keyword matches to semantic matches on a conceptual level can help identify relevant articles, even when different terms are used, thereby aiding users in discovering publications from subfields they are unfamiliar with. Complementing this, knowledge graphs (KGs) have established themselves as a versatile data structure for representing semantic relationships between interconnected entities like institutions, authors, topics, research fields, and other concepts.

Two popular examples of platforms that have incorporated KGs are Microsoft Academic (Wang et al., 2020a) and Semantic Scholar (Kinney et al., 2023). Microsoft Academic created the Microsoft Academic Graph, which supports semantic search, contextual query understanding, and personal recommendations. Similarly, the Semantic Scholar platform operates on the Semantic Scholar Academic Graph with more than 200 million papers. While these platforms offer a range of graph-based features and visualizations, they introduce usability hurdles by rendering graphical search interfaces more complex. Graphical interfaces can become less effective for exploratory search because of the added layers of complexity, causing users to experience cognitive overload (Sweller, 1988). This might be exemplified by the decline and eventual termination of Microsoft Academic in 2021, whose intricate interface likely has contributed to deterring users (Orduña Malea et al., 2014).

To address the complexity of graphical semantic search interfaces, we propose developing a conversational interface for discovering scholarly publications via dialogue interactions, leveraging a KG data structure. The emerging paradigm of conversational search promises to satisfy information needs using intuitive information-providing conversations while avoiding information overload (Radlinski and Craswell, 2017). Through interactions with conversational agents, users can resolve ambiguities, refine their queries, narrow down the relevant search space, and extract novel insights. Our study aims to provide insights into how conversational search systems integrated with KGs can enhance the discovery of publications, thereby improving navigation and information retrieval in the scholarly research landscape. To demonstrate the effectiveness of our developed system, we utilize the open-source corpus of the ACL Anthology as our data foundation. The source code, models, datasets, and questionnaires are made available via a public GitHub repository.[1] Our three main contributions are as follows: (1) We propose an architecture for integrating a conversational exploratory search system with a scholarly KG. (2) We implement the system by assembling different task-specific language models. (3) We conduct both a model-centric performance assessment and a human evaluation of the developed system with 40 participants.

2 Related Work

Conversational search systems are defined as conversational interfaces that support acquiring information through multi-turn dialogues. These systems progressed significantly in recent years, largely driven by the rapid adoption of large language models (LLMs). A growing body of research focuses on augmenting conversational search systems with LLMs (Schneider et al., 2024c), including utterance understanding (Kuhn et al., 2023), dialogue management (Friedman et al., 2023), knowledge retrieval (Lewis et al., 2020), and response generation (Sekulic et al., 2024; Schneider et al., 2024b). While LLMs hold great potential for conversational search systems, they are not without shortcomings. LLMs can hallucinate or omit crucial information, and their outputs often lack transparency regarding the source of generated content (Ji et al., 2023). In addition, LLMs are usually non-deterministic, posing challenges in ensuring consistent and correct knowledge due to the randomness in their text generation processes.

To mitigate issues of factuality and reliability in conversational systems and LLMs, researchers have studied using KGs to ground outputs in verifiable data sources. Integrating KGs with dialogue systems has long been a focus in the literature. KGs can replace static domain knowledge with dynamic ontologies and have shown effectiveness in conversational question answering (QA) (Christmann et al., 2019; Schneider et al., 2024a). By navigating entity nodes and relationships, KGs enhance conversational context and information exploration. Numerous studies support the use of KGs for improving utterance understanding, response generation, and dialogue management (Chen et al., 2019, 2023). While KGs are increasingly being combined with conversational agents in fields such as healthcare, law, and business, there remains a significant gap in their application within the scholarly domain. Thus far, Meloni et al. (2023b) are the only ones to propose combining a conversational agent with the *Academia/Industry DynAmics* (AIDA) KG (Angioni et al., 2021). Their AIDA chatbot focuses on QA by executing database queries to count, list, compare, or describe scholarly entities (e.g., authors or conferences), thereby offering senior researchers an overview of the research landscape through bibliometric data (Meloni et al., 2023a).

In contrast, our proposed system supports the conversational discovery of research articles for users with vague goals in open-ended search scenarios, building on insights from our previous work (Schneider et al., 2023b). Therefore, unlike the AIDA chatbot, which primarily assists senior researchers, our proposed system is designed to support exploratory information search for non-expert users looking to discover relevant publications on a given topic without requiring in-depth knowledge.

---

[1] Repository: github.com/philotron/CS-Scholarly-KG

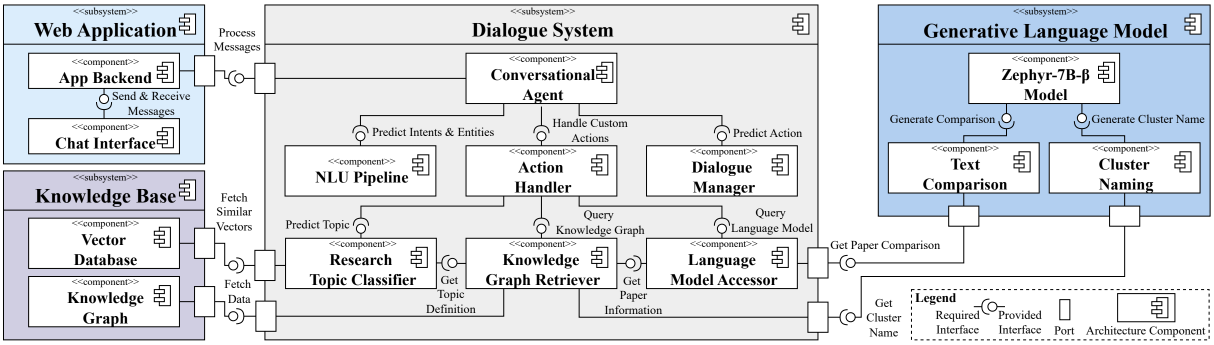


Figure 1: Architectural components of the conversational exploratory search system.

3 Conversational Search System

3.1 System Architecture

The developed dialogue system helps users narrow down relevant publications via a three-phase search process. An overview of the conversation flow is illustrated in Figure 4 in Appendix A. In the first phase, the system receives a short description of a search goal (e.g., “I want to study how people express their feelings on social media.”) and then it assists users by recommending an appropriate NLP research topic to explore (e.g., *Emotion Analysis*). Second, users can iteratively choose thematic clusters of articles within the selected research topic (e.g., *Emotion Detection in Social Media Text*). Third, users are presented with a list of articles at the lowest cluster level, where they can compare papers based on summarized information from the abstracts. Finally, users have the option to either access links to the full texts of the papers or continue exploring other research topics, clusters, or articles. In Section 4.1, we will elaborate on the developed NLP models powering the three-phase search process through (1) topic classification, (2) text clustering, as well as (3) text summarization.

Figure 1 illustrates the system architecture, which is structured into four distinct subsystems. Each subsystem encompasses multiple components responsible for typical dialogue system functions. The front end is a conversational interface implemented as a web application subsystem using the open-source framework Streamlit.[2] It features a basic chat interface with a text input form and a scrollbar for the current dialogue history. User messages entered in the chat interface are sent to a dialogue system built with RASA, an open-source machine learning framework.[3] RASA supports the development of conversational agents with a natural language understanding (NLU) pipeline that recognizes intents and entities from user utterances. Based on these semantically parsed user utterances, the agent’s dialogue manager, which takes into account dialogue states, dialogue policies, and conversation context, predicts the system’s next actions. Aside from standard actions like producing a simple response, the agent connects with an action handler component to implement custom actions. One custom action is the KG retriever component. It enables the construction and execution of structured queries to retrieve data from the scholarly KG, such as abstracts, thematic clusters, or research topics. It connects with the knowledge base subsystem, which hosts the KG in a Neo4j property graph database.[4] For the query construction, extracted entities from user utterances are matched with those existing in the KG to fill out template queries.

In addition to the KG, the knowledge base subsystem hosts the open-source vector database Weaviate, which performs embedding-based similarity search.[5] Together with the KG, the vector database supports the research topic classifier component by finding the closest research topics from user inquiries. Another component that powers a custom action is the language model accessor, which provides a connecting endpoint to the generative language model subsystem. Inside this subsystem, we host the open-source LLM Zephyr-7B-Beta (Tunstall et al., 2023). The subsystem offers two inference endpoints for dynamic prompting. The endpoints are used to generate names for paper clusters and summarized paper comparisons.

The described system is deployed on three virtual machines (VMs) in a cloud environment. The first VM operates the dialogue system that interacts

---

[2] Streamlit: https://streamlit.io

[3] RASA: https://rasa.com

[4] Neo4j: https://neo4j.com

[5] Weaviate: https://weaviate.io

directly with users through the conversational interface. The second VM acts as a database server, while the third VM, equipped with a GPU (16 GB memory), hosts the large language model. Despite the architecture's various technical components depicted in Figure 1, the conversational interface hides the complexity of the underlying KG, providing a highly accessible search experience.

3.2 Knowledge Graph and Vector Database

To establish the data foundation for the conversational search system, we constructed a domain-specific KG with over 85,000 research articles sourced from the ACL Anthology.[6] A compact overview of the data schema with entity nodes and relations is presented in Figure 5 in Appendix A.

Sourcing articles from the ACL Anthology provided us with detailed metadata on authors, venues, and publication years that were automatically transformed into nodes and relations of the KG. In addition, we assigned each article to one or multiple research topics. To achieve this, we used a previously established taxonomy of NLP research topics from Schopf et al. (2023), which is organized as a two-level hierarchy with main topics and subtopics, along with topic names and human-written definitions. This taxonomy includes 12 main topics, such as *Text Generation* or *Sentiment Analysis*, and a total of 71 subtopics (e.g., *Question Generation* or *Emotion Analysis*). For classifying articles, we employed two fine-tuned language models for classifying publications: a SPECTER2-based model (Singh et al., 2023) for multi-label topic classification based on the used NLP taxonomy (Schopf et al., 2023; Schopf and Matthes, 2024), and a SciNCL-based model (Ostendorff et al., 2022) to classify if a publication is a survey paper consolidating information from several other publications. The taxonomy inside the KG is later applied to train a classification model that predicts a relevant NLP subtopic based on a described search goal. By providing topic definitions and listing related topics, the conversational agent can guide users through the NLP taxonomy.

While effective for broad classification, the two-level NLP taxonomy is not granular enough to account for thematic differences within a given subtopic. For example, the subtopic *Emotion Analysis* includes over 780 publications, which share only a few common characteristics. To have a more fine-grained filtering mechanism, we clustered papers based on their title and abstract content (e.g., specific techniques, application domains, or benchmark datasets). These thematic clusters are precomputed and modeled as nodes in the KG. The clustering and cluster naming methods will be discussed in more detail in Section 4.1.

In addition to the KG database, we installed a vector database that supports various embedding models and similarity metrics, making it ideal for efficiently ranking semantically similar documents. We employed the SPECTER2 embedding model (Singh et al., 2023) for generating vectors from papers' titles and abstracts and used them for the mentioned research topic classification, mapping NLP topics from the taxonomy to user requests during the search dialogue in real-time. A document identifier in the vector database links these embeddings to the papers in the KG. As a last construction step, we further enriched the KG with metadata from the Semantic Scholar API, including one-sentence *too long; didn't read* (TLDR) summaries, citation counts, and publication references.

4 Results and Discussion

4.1 Model Training and Evaluation

Research Topic Classification. In the following sections, we report the results of training and evaluating the NLP models that underpin the three-phase search process of our developed dialogue system: (1) research topic classification, (2) article text clustering, and (3) comparative article summarization. The first phase involves classifying an uttered search goal or problem description into a fitting NLP research topic. This is especially helpful for users in exploratory search settings because they may not be familiar with all existing fields of study and struggle to phrase their queries using the correct terminology. Due to the absence of datasets that map search goals expressed in layman's terms to NLP topics, we created a synthetic multi-class dataset using GPT-3.5-Turbo (version: 0613) (OpenAI, 2022). We prompted the LLM to generate questions on the 12 main topics in our taxonomy using three distinct personas: a computer science student with only peripheral NLP knowledge, a businessperson with practical experience of NLP tools but minimal technical expertise, and a non-technical, non-academic individual whose technology use is limited to basic tasks. Persona-specific prompting yielded diverse inquiries in lay-

---

[6] ACL Anthology: https://aclanthology.org

man's language. For example, the question "How are computers able to respond when we ask them questions?" was generated for the topic *Natural Language Interfaces*. To account for questions unrelated to NLP, we also generated a set of out-of-scope questions such as "Who discovered the laws of thermodynamics?" Following a quality inspection of the synthetically produced questions, we assembled a training dataset of 1601 examples, consisting of 120 questions for each of the 12 topics and 161 general questions. We also derived a test dataset containing 364 examples with a balanced class distribution similar to the training dataset.

In our experiments, we evaluated three classification approaches: vector similarity search, prompting a LLM (GPT-3.5-Turbo), and few-shot fine-tuning of a transformer model with the SetFit framework (Tunstall et al., 2022). Concerning the vector search approach with the SPECTER2 model, we measured the cosine similarity to compare vectors of embedded user queries with paper embeddings in our vector database to retrieve the 100 most similar papers. We found that a similarity threshold below 77% effectively filters out the non-NLP-related questions. Using the scholarly KG, we aggregated linked topics for these papers and predicted the most frequent topic as output class. For the LLM approach, we crafted a zero-shot prompt for GPT-3.5-Turbo, provided in Appendix A, which instructed the LLM to classify the appropriate topic from the list of 12 main topics or answer with "None" if the question was not related to NLP. Moreover, we tested the SetFit approach for fine-tuning the sentence transformer model multi-qa-MiniLM-L6-cos-v1 (Wang et al., 2020b). We trained for 3 epochs, a batch size of 16, and 30 SetFit iterations for contrastive learning.

Figure 2 illustrates the classification performance for each approach. While vector search achieved a macro F1-score below 0.50., GPT-3.5-Turbo achieved a score near 0.75; however, it exhibited a bias toward particular topics, leading to overprediction and incorrectly classifying general questions as NLP topics. The SetFit model demonstrated superior performance over the two other approaches with a score of 0.95. Consequently, we implemented the topic classifier component with this fine-tuned model for main topic classification in combination with similarity search for classifying the subtopic. This allows a more nuanced classification of user queries into subtopics, given the more detailed information in the paper abstracts.

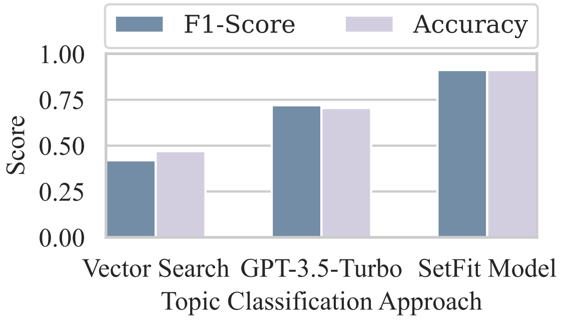


Figure 2: Comparison of accuracy and F1-scores for three topic classification approaches.

Article Text Clustering. After selecting an NLP subtopic, the conversational search agent guides users by presenting clusters of papers to further narrow down the search space. We tried out various clustering methods for thematically grouping similar papers because listing all papers within one subtopic at once is impractical. Selecting thematic clusters makes it easier for users to find relevant papers by iteratively choosing smaller, more specific clusters. Our experiments indicated that agglomerative clustering, a hierarchical bottom-up clustering approach, was the most effective in producing mutually exclusive clusters at each hierarchy level (Murtagh and Contreras, 2012). We employed the SPECTER2 embeddings of the publications, which were the same ones used for the similarity search as part of the research topic classification.

Initially, we used a distance threshold of 10, resulting in clusters averaging 15 publications each. However, this led to too many clusters inside a given research topic, making cluster selection very cumbersome. The distance threshold represents the maximum distance within which items are clustered together. To improve the clustering, we adopted an iterative hierarchical approach. We progressively decreased the distance threshold at each cluster level, keeping the number of clusters small while increasing the paper similarity within each subcluster to facilitate user navigation. Clustering stopped when fewer than 10 publications remained per cluster, ensuring a user-friendly number to display. Overall, we constructed a granular hierarchy of 47,035 thematic clusters, which were modeled as nodes in the constructed scholarly KG.

Next, it was necessary to assign human-readable cluster names to help users identify relevant clusters during the conversational search interaction. We applied a term frequency-inverse document frequency (TFIDF) vectorizer to extract important words from the titles of all publications within each cluster. Through several experiments, we found that an n-gram range of (2,5), considering sequences of two to five consecutive words, yielded good cluster names. However, we observed that a few clusters had identical names. To resolve this, we performed another cluster naming iteration, taking into account all previous names. If a name was repeating, we selected the second or third most relevant TFIDF label to ensure unique names. While the TFIDF-derived names were readable, they often contained too much detail and domain-specific words, rendering them less accessible to non-expert users. To make cluster names more understandable, the aforementioned Zephyr-7B-Beta LLM was applied. Within the LLM subsystem in our architecture, a specialized component was developed for cluster naming. A dynamic prompt, detailed in the GitHub repository, was created to transform an existing TFIDF cluster name alongside five randomly selected paper titles from a chosen cluster into a more comprehensible cluster name (e.g., *Emotion Detection in Social Media Text* or *Extraction of Concept Maps for Multi-Document Summarization*). To minimize response latency during a conversation, names for clusters and subclusters were pre-computed and stored in the scholarly KG.

Comparative Text Summarization. In the last phase of the conversational search process, users can compare papers listed at the lowest cluster level. Although these papers are already thematically related, the comparison allows users to discern specific similarities and differences, aiding in determining which paper to read more thoroughly. The language model subsystem allows for the summarization of objectives and results of two selected papers, which are generated in real-time upon request with Zephyr-7B-Beta. Given that injecting full abstracts can impede the LLM's ability to accurately detect objectives and results, only relevant portions of the abstract are provided in a dynamic prompt, which has been shown to reduce hallucinated outputs (Martino et al., 2023).

To this end, we first classified abstract sentences that discuss objectives or results using SciBERT, a language model pre-trained on scientific text (Beltagy et al., 2019). We fine-tuned SciBERT on a labeled dataset from Gonçalves et al. (2020), including 500 computer science abstracts and 3,287 sentences classified as background, methods, objectives, results, or conclusions. After hyperparameter optimization, our fine-tuned model achieved an F1-score of 75.39%, which is around one percentage point higher than the model from Gonçalves et al. (2020) with 74.60%. Finally, we applied our model to all the publication abstracts in our KG and stored the classified objectives and results sentences accordingly. More details about our fine-tuned model are available in the repository. A dynamic LLM prompt for text summarization was crafted, as shown in Table 5 in Appendix A. Two researchers manually assessed the generated comparisons. Initial experiments with other models, such as Falcon-7B and Llama-2-7B, showed that these models were less attuned to following the instruction, often producing hallucinated content or excessively verbose responses, making them unsuitable for conversational interactions. As a result, we selected Zephyr-7B-Beta, which delivered better output in terms of style and faithful content.

4.2 Human Evaluation

Experiment Design. To evaluate the three-phase search system in an end-to-end manner, we designed a user study in which participants explored publications related to two predefined search scenarios. They interacted with the conversational search interface and a graphical interface featuring a traditional text-based search, allowing us to compare the effectiveness of both systems. For the experiment, we recruited 40 participants from university courses and social networks according to criteria that match our target user group of non-experts. Table 3 in Appendix A gives an overview of participant demographics. All participants had at least basic technical knowledge, good English proficiency, and an interest in NLP without having expert-level knowledge. The gender composition was 35% female and 65% male, ranging in age from 20 to 29 years, with an average age of 25.

Prior to the user experiment, we randomly assigned participants into two groups (Group A and Group B), which determined the sequence in which each group used the search interfaces for the two search scenarios (Scenario 1 and Scenario 2). We ensured that the demographic characteristics of both groups were similarly distributed. Group A is exposed to Scenario 1 with the conversational interface first, followed by Scenario 2 with the graphical interface. Conversely, Group B is exposed to Scenario 1 with the graphical interface first, then Scenario 2 with the conversational interface. This

Evaluation Metric **Mean (Std. Dev.)**	**Scenario 1**		**Scenario 2**	
	Conversational	**Graphical**	**Conversational**	**Graphical**
System usability scale	76.00 (18.94)	77.25 (15.28)	76.63 (16.63)	65.25 (23.91)
Readability	4.50 (0.95)	3.40 (1.14)	4.45 (0.76)	3.20 (1.54)
Correctness	4.25 (0.97)	4.05 (1.00)	4.25 (0.72)	3.85 (1.31)
Usefulness	4.50 (0.61)	3.65 (0.99)	4.30 (0.80)	2.95 (1.23)
Summary quality	4.10 (0.85)	-	4.15 (0.67)	-
Overall satisfaction	4.15 (0.88)	3.45 (1.00)	4.10 (1.07)	2.85 (1.14)

Table 1: Overview of mean and standard deviation of evaluation metrics by search scenario and interface type.

crossover design allows each participant to test both interfaces but for a different scenario to avoid learning effects. Scenario 1 is about analyzing emotional expressions on social media related to mental health during the COVID-19 pandemic, while Scenario 2 focuses on creating multiple-choice exams for a programming course. Both scenarios end with the instruction: “Your task is to use the provided search interface below to find papers related to the described scenario.” The full scenario descriptions are documented in Table 2 in Appendix A.

Participants were given approximately 10 minutes to interact with the interfaces (see screenshots in Table 4), followed by an evaluation questionnaire, which we share in the repository. The latter includes 10 questions from the *system usability scale* (SUS) (Brooke, 1996). The SUS metric is calculated using a specific formula, resulting in a value between 0 and 100, with 68 being considered average. Furthermore, participants were asked five questions to rate the general information quality in terms of readability, correctness, and usefulness, the quality of the generated comparisons, as well as overall satisfaction. The questions were answered on a 5-point Likert scale, where a rating of 5 denoted the most favorable value. In addition, we included two open-ended free-text fields for feedback on the system’s strengths and weaknesses.

Quantitative Analysis of Evaluation Metrics. Based on the questionnaire responses, we conducted both quantitative and qualitative analyses. Table 1 lists the mean and standard deviation for each evaluation metric grouped by scenario and interface. We found that all data points were within reasonable ranges without containing significant outliers. Generally, the ratings for the conversational interface tend to be more favorable across the various evaluation metrics. This is also reflected in the overall satisfaction scores for Scenario 1 and Scenario 2, with ratings of 4.15 and 4.10 for the conversational interface compared to 3.45 and 2.85 for the graphical interface. The conversational interface especially surpasses the graphical interface in readability and usefulness metrics, as reflected by mean ratings that were around one point higher across both scenarios. We hypothesize that the dialogue interaction was not as overwhelming, delivering information in a more digestible format and increasing its overall utility by offering additional choices for paper selection. This positive feedback was likely also influenced by the quality of summarized paper comparisons, which achieved solid scores of 4.10 in Scenario 1 and 4.15 in Scenario 2.

Inspecting the perceived system usability, the SUS scores for Scenario 1 were similar for both interfaces, with SUS scores at 76.00 and 77.25. Since all participants presumably had prior experience on how to use a text-based search engine as well as a standard chat interface, it is not surprising that the scores are similar. It is very likely that each participant was already acquainted with operating both types of interfaces. In Scenario 2, the conversational interface maintained a comparable score of 76.63, whereas the graphical interface had a lower score of 65.25, suggesting that the conversational interface offers more consistent usability across different search scenarios. This observation is corroborated by the rating distributions illustrated in Figure 3. The latter shows that Scenario 2 received worse ratings compared to Scenario 1 for both evaluated interfaces, with the rating distributions shifting towards the lower scores, possibly indicating that Scenario 2 was slightly more difficult with regard to discovering relevant papers. This effect was especially pronounced for the graphical interface compared to the more consistent performance of the conversational interface. The more stable ratings of the conversational interface could suggest it retains usability and information relevancy, even during more challenging exploration tasks.

Qualitative Analysis of Participant Feedback. Our qualitative analysis of the free-text responses concerning the systems’ strengths and weaknesses confirmed the quantitative results, revealing that

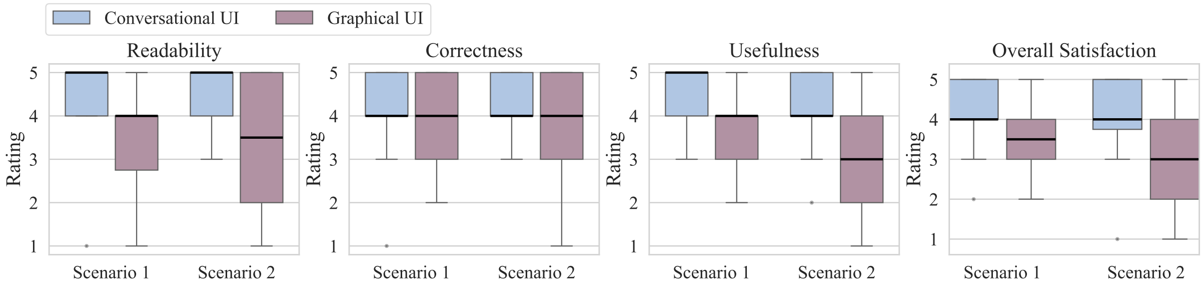


Figure 3: Comparison of rating distribution between the conversational and graphical user interface (UI) for readability, correctness, usefulness, and overall satisfaction. The thick line intersecting the box marks the median.

the conversational interface received higher ratings and demonstrated more consistent usability than the graphical interface. Every participant was assigned an anonymous identifier between P1 and P40 to protect their privacy. In the following paragraphs, cited questionnaire responses are presented in quotation marks and assigned to their identifiers.

The first notable strength of the conversational search interface, mentioned in nearly every second feedback comment, was the system's ease of use. For example, participant P9 remarked, "It is easy to use and to find topics & papers even if the prior exposure to the given topic is low." Users appreciated that they could immediately start talking with the conversational agent without requiring extensive knowledge of the interface or the search domain, unlike many graphical interfaces. A second strength highlighted by users was the system's guidance and structured navigation abilities, with one participant positively noting the "Step by step process to narrow down the search and avoid search results that are not related to your query" (P29). This feature effectively addressed the search-related vocabulary mismatches, as exemplified by the comment: "I don't need to know exact terms or what im looking for" (P40). Lastly, users valued the time-saving clustering and summarization features, which helped them avoid going through individual abstracts from long lists of papers. As participant P11 stated, "[...] it understands the content of the paper and can aggregate it, without me having to manually go into the files to read the Abstract." These findings suggest that conversational agents can help alleviate problems associated with cognitive overload (Sweller, 1988) by gradually communicating condensed information.

Yet, a couple of participants initially struggled with understanding the three-phase search flow (e.g., "In the first a few minutes it's hard to understand what I can reach at the end of conversation" (P35)). Some were also confused by the two options of selecting the suggested user response inputs displayed as buttons versus entering free-form text. This was especially the case when they wanted to reverse a choice, which participant P20 remarked, "The options to backtracking are a bit unclear at first." Other participants expressed a desire to "converse more freely" (P9), similar to those offered by general-purpose LLMs. Strengthening the integration of LLMs could accommodate this preference, as LLMs excel in contextual understanding of queries and navigating complex conversation logic more effectively than intent-based dialogue systems. We observed that certain users attempted to input very long requests or copy-pasting problem descriptions, an interaction more akin to LLM services like ChatGPT (OpenAI, 2022), where users input a prompt, check the output, and refine the prompt without engaging in a proper dialogue. This type of interaction does not align with how our task-oriented dialogue system was designed to operate. Nonetheless, the evaluation shows that nearly all participants quickly figured out how our conversational system works after a few dialogue turns.

5 Conclusion

We proposed a conversational exploratory search system integrated with a scholarly KG. Our study details the architectural components and presents results from training and evaluating language models that underpin the three-phase search process, including research topic classification, text clustering, and text summarization. We conducted a human evaluation to assess the system's effectiveness, identifying its perceived strengths and potential improvements. Our findings offer practical insights into the design and implementation of conversational search systems for the scholarly domain.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) Software Campus grant 01IS17049.

References

Simone Angioni, Angelo Salatino, Francesco Osborne, Diego Reforgiato Recupero, and Enrico Motta. 2021. Aida: A knowledge graph about research dynamics in academia and industry. *Quantitative Science Studies*, 2(4):1356–1398.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on EMNLP and the 9th IJCNLP*, pages 3615–3620, Hong Kong, China. ACL.

Lutz Bornmann, Robin Haunschild, and Ruediger Mutz. 2021. Growth rates of modern science: a latent piecewise growth curve approach to model publication numbers from established and new literature databases. *Humanities and Social Sciences Communications*, 8.

John B. Brooke. 1996. Sus: A 'quick and dirty' usability scale. *Usability evaluation in industry*, 189(194):4–7.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Conference on EMNLP and the IJCNLP*, pages 1803–1813, Hong Kong, China. ACL.

Zheng Chen, Ziyan Jiang, Fan Yang, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Aram Galstyan. 2023. Graph meets LLM: A novel approach to collaborative filtering for robust conversational understanding. In *Proceedings of the 2023 Conference on EMNLP: Industry Track*, pages 811–819, Singapore. ACL.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 729–738.

Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint*.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Sérgio Gonçalves, Paulo Cortez, and Sérgio Moro. 2020. A deep learning classifier for sentence classification in biomedical and computer science abstracts. *Neural Computing and Applications*, 32(11):6793–6807.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. 2023. The semantic scholar open data platform. *arXiv preprint*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Clam: Selective clarification for ambiguous questions with generative language models. In *ICML 2023 Workshop on Deployment Challenges for Generative AI*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 33:9459–9474.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer.

Antonello Meloni, Simone Angioni, Angelo Salatino, Francesco Osborne, Aliaksandr Birukou, Diego Reforgiato Recupero, and Enrico Motta. 2023a. Aida-bot 2.0: Enhancing conversational agents with knowledge graphs for analysing the research landscape. In *International Semantic Web Conference*, pages 400–418. Springer.

Antonello Meloni, Simone Angioni, Angelo Salatino, Francesco Osborne, Diego Reforgiato Recupero, and Enrico Motta. 2023b. Integrating conversational agents and knowledge graphs within the scholarly domain. *IEEE Access*, 11:22468–22489.

Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

Enrique Orduña Malea, Alberto Martín-Martín, Juan M Ayllón, and Emilio Delgado-López-Cózar. 2014. The silent fading of an academic search engine: the case of microsoft academic search. *Online Information Review*, 38(7):936–953.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on EMNLP*, pages 11670–11688, Abu Dhabi, United Arab Emirates. ACL.

Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 117–126, New York, NY, USA. ACM.

Phillip Schneider, Anum Afzal, Juraj Vladika, Daniel Braun, and Florian Matthes. 2023a. Investigating conversational search behavior for domain exploration. In *Proceedings of the 45th European Conference on Information Retrieval (ECIR)*, pages 608–616, Dublin, Ireland. Springer Nature Switzerland.

Phillip Schneider, Manuel Klettner, Kristiina Jokinen, Elena Simperl, and Florian Matthes. 2024a. Evaluating large language models in semantic parsing for conversational question answering over knowledge graphs. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 807–814, Rome, Italy. SciTePress.

Phillip Schneider, Manuel Klettner, Elena Simperl, and Florian Matthes. 2024b. A comparative analysis of conversational large language models in knowledge-based text generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 358–367, St. Julian's, Malta. Association for Computational Linguistics.

Phillip Schneider, Wessel Poelman, Michael Rovatsos, and Florian Matthes. 2024c. Engineering conversational search systems: A review of applications, architectures, and functional components. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI)*, pages 73–88, Bangkok, Thailand. Association for Computational Linguistics.

Phillip Schneider, Nils Rehtanz, Kristiina Jokinen, and Florian Matthes. 2023b. From data to dialogue: Leveraging the structure of knowledge graphs for conversational exploratory search. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 609–619, Hong Kong, China. Association for Computational Linguistics.

Tim Schopf, Karim Arabi, and Florian Matthes. 2023. Exploring the landscape of natural language processing research. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1034–1045, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tim Schopf and Florian Matthes. 2024. Nlp-kg: A system for exploratory search of scientific literature in natural language processing. In *Proceedings of the 62st Annual Meeting of the ACL (Volume 3: System Demonstrations)*, Bangkok, Thailand. ACL.

Ivan Sekulic, Krisztian Balog, and Fabio Crestani. 2024. Towards self-contained answers: Entity-based answer rewriting in conversational search. In *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval*, CHIIR '24, page 209–218, New York, NY, USA. ACM.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on EMNLP*, pages 5548–5566, Singapore. ACL.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. In *NeurIPS 2022 Workshop on Efficient Natural Language and Speech Processing*.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020a. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*, volume 33, pages 5776–5788.

A Appendix

The Appendix provides further insights into the results of our research, including a finite-state diagram of the conversational search flow (Figure 4), the semantic data model of the scholarly KG (Figure 5), the scenario descriptions for the human evaluation (Table 2), a tabular overview of the participant demographics (Table 3), screenshots of the conversational and graphical interface (Table 4), and a collection of the crafted LLM prompts (Table 5).

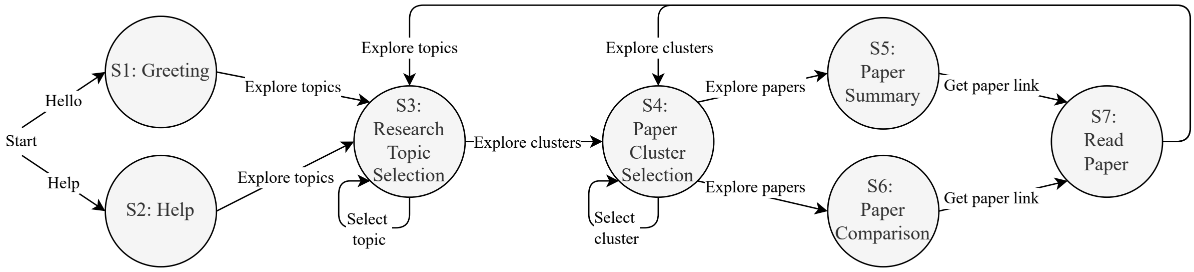


Figure 4: Conversational search flow illustrated as dialogue states (S1-S7). The three-phase search process encompasses: first, identifying a research topic (S3); second, choosing clusters of publications (S4); and third, comparing publications via short summaries (S5-S6).

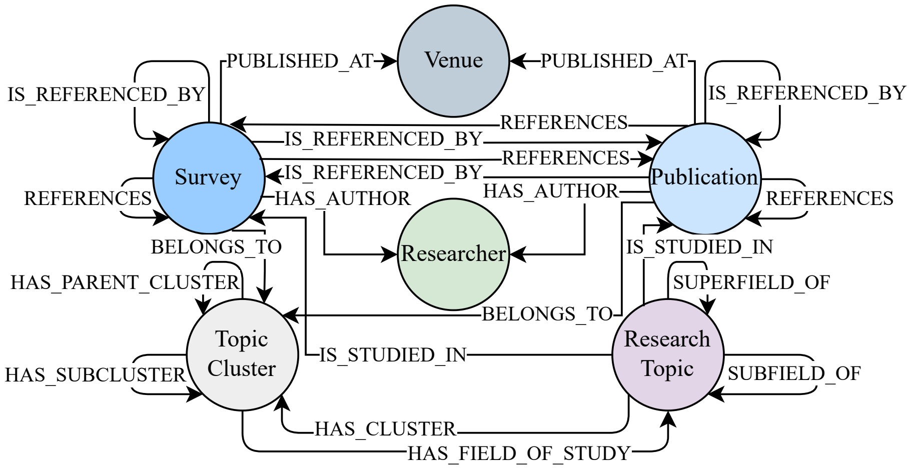


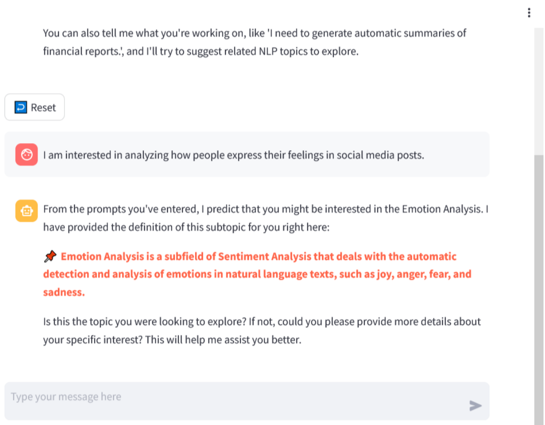
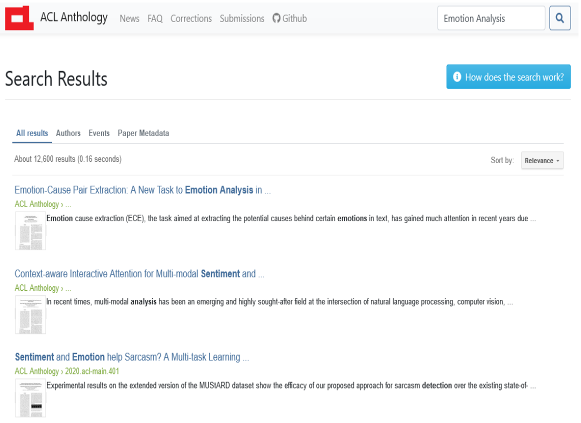
Figure 5: Semantic data model of the scholarly knowledge graph.

Description of Scenario 1
Imagine you are interested in mental health and emotional changes during the COVID pandemic. You want to analyze how people express their feelings on social media platforms during the pandemic. Your goal is to study their emotions to learn how they handle stress and anxiety. To deal with the enormous volume of data available online, you are looking for ways to automate the analysis process using NLP techniques. Your task is to use the provided search interface below to discover papers related to the described scenario. You have up to 8 minutes for your exploratory search. You are encouraged to "think out loud". Afterward, you will fill out an evaluation questionnaire to provide feedback on your search experience.
Description of Scenario 1
Imagine you are an instructor teaching a programming course in Python. You want to ensure that the exam questions reflect the learning progress of the students. Your goal is to generate multiple-choice exams according to their knowledge level. To achieve this, you are looking for ways to automatically create exam questions based on the course materials using NLP techniques. Your task is to use the provided search interface below to discover papers related to the described scenario. You have up to 8 minutes for your exploratory search. You are encouraged to "think out loud". Afterward, you will fill out an evaluation questionnaire to provide feedback on your search experience.

Table 2: Scenario descriptions of the exploratory search task for the human evaluation.

Demographic Variable	**Group A ($n = 20$)**	**Group B ($n = 20$)**	**Overall ($n = 40$)**
Mean age (age range)	25.10 (20 to 29)	24.95 (23 to 28)	25.03 (20 to 29)
Male	13	13	26
Female	7	7	14
High school degree	1	-	1
Bachelor's degree	14	17	31
Master's degree	5	3	8
No NLP knowledge	1	2	3
Beginner NLP knowledge	15	13	28
Advanced NLP knowledge	4	5	9
English CEFR level B1 or B2	3	5	8
English CEFR level C1 or C2	17	15	32

Table 3: Overview of study participant demographics.

Conversational Interface	**Graphical Interface**
	

You can also tell me what you're working on, like 'I need to generate automatic summaries of financial reports.', and I'll try to suggest related NLP topics to explore.

I am interested in analyzing how people express their feelings in social media posts.

From the prompts you've entered, I predict that you might be interested in the Emotion Analysis. I have provided the definition of this subtopic for you right here:

📌 **Emotion Analysis is a subfield of Sentiment Analysis that deals with the automatic detection and analysis of emotions in natural language texts, such as joy, anger, fear, and sadness.**

Is this the topic you were looking to explore? If not, could you please provide more details about your specific interest? This will help me assist you better.

Emotion Analysis

Search Results

About 12,600 results (0.16 seconds)

Sort by: Relevance

Emotion-Cause Pair Extraction: A New Task to **Emotion Analysis** in ...
ACL Anthology › ...
Emotion cause extraction (ECE), the task aimed at extracting the potential causes behind certain **emotions** in text, has gained much attention in recent years due ...

Context-aware Interactive Attention for Multi-modal **Sentiment** and ...
ACL Anthology › ...
In recent times, multi-modal **analysis** has been an emerging and highly sought-after field at the intersection of natural language processing, computer vision, ...

Sentiment and **Emotion** help Sarcasm? A Multi-task Learning ...
ACL Anthology › 2020.acl-main.401
Experimental results on the extended version of the MUStARD dataset show the efficacy of our proposed approach for sarcasm **detection** over the existing state-of- ...

Table 4: Visual side-by-side comparison of the conversational and graphical interface from the human evaluation.

Prompt 1: Cluster Name Generation (Zephyr-7B-Beta)
Considering the themes and topics from the following TFIDF cluster tag: “{tfidf_cluster_name}”, please provide a concise and descriptive name for a cluster that includes these {len(paper_list)} academic papers: {paper_titles_formatted} Respond with just the cluster name, based on the overarching themes evident in the titles and the TFIDF tag. Don’t include the original TFIDF cluster tag and the word “Cluster” in your response.
Prompt 2: Comparative Text Summarization (Zephyr-7B-Beta)
Please provide a comparative analysis of the objectives of two scientific papers. Refer the papers with their real ids: Paper {id_a}’s objective is: {obj1} Paper {id_b}’s objective is: {obj2} Highlight the key differences and similarities between Paper {id_a} and Paper {id_b}. Use simple language. Please provide a comparative analysis of the results of two scientific papers.: Refer the papers with their real ids: Results of Paper {id_a}: {res1} Results of Paper {id_b}: {res2} Highlight the key differences and similarities between Paper {id_a} and Paper {id_b}. Use simple language. Please provide a comparative analysis of the TLDR of two scientific papers.: TLDR of Paper {id_a}: {tldr1} TLDR of Paper {id_b}: {tldr2} Highlight the key differences and similarities between Paper {id_a} and Paper {id_b}. Use simple language.
Prompt 3: LLM-Based Research Topic Classification (GPT-3.5-Turbo)
You are supposed to classify a query into one of the topics provided. These topics are various fields of NLP. Your answer should be in the following format: *topic name* Nothing else should be included in the output. Make sure there is no extra punctuation including full stops, quotation marks or anything of that sort. You are supposed to EXACTLY use the topics from the list provided. If you think it is a random question and not in the field of NLP, then return the topic as “none”. You can only provide your answer from the following topics and the topics are: Multimodality Natural Language Interfaces Semantic Text Processing Semantic Analysis Syntactic Text Processing Linguistic and Cognitive NLP Responsible NLP Reasoning Multilinguality Information Retrieval Information Extraction and Text Mining Text Generation Query: {query}. Topic:

Table 5: Overview of large language model prompts for various generative tasks using Zephyr-7B-Beta and GPT-3.5-Turbo. Dynamically inserted variables are enclosed within curly brackets.

A Hybrid Retrieval Approach for Advancing Retrieval-Augmented Generation Systems

Nguyen Nam Doan[1], **Aki Harma**[1], **Remzi Celebi**[1],**Valeria Gottardo**[2]
[1]Maastricht University, The Netherlands
[2]NLC Health Ventures, The Netherlands
{nam.doan, aki.harma, remzi.celebi}@maastrichtuniversity.nl
valeria.gottardo@nlc.health

Abstract

Retrieval-Augmented Generation (RAG) has become a promising solution for utilizing Large Language Models (LLMs) in domain-intensive question-answering tasks. The performance of RAG is greatly influenced by the retriever component, which typically relies on semantic similarity between the text embeddings of the query and the passages to identify the relevant context. However, text embedding models may only capture the semantic meaning of individual passages, potentially neglecting global relationships between them. To address this limitation, we propose a hybrid retrieval method that integrates embeddings encoded from textual and knowledge graph information. Although in this paper, the knowledge graphs describe the passage relationships in a health-tech industry use case, the hybrid embedding solution is designed to be generic. Furthermore, the proposed retrieval approach aims to offer straightforward implementation without requiring complex joint learning processes. Our results on custom test sets demonstrate significantly enhanced accuracy and ranking of the retriever, thus, supporting the LLM-based reader component in generating more accurate responses.

1 Introduction

The use of pre-trained Large Language Models (LLMs) has gained considerable attention for specific question-answering (QA) tasks, due to their ability to understand and generate natural language (De Angelis et al., 2023). This capability results from their extensive pre-training on diverse text datasets and a large number of parameters, which endows them with the ability to memorize and utilize learned knowledge (Roberts et al., 2020; Petroni et al., 2019). However, implementing pre-trained services within corporate settings faces certain challenges. One notable limitation is the inability to generate accurate and faithful responses for questions related to specific domains and business use cases, often referred to as "hallucination". This constraint stems from knowledge boundaries, which include a lack of specialized domain knowledge and the absence of up-to-date information in the pre-trained data (Huang et al., 2023).

Fine-tuning generative LLMs with a target domain dataset has been proposed as a solution to this problem. This approach adapts the models for specific tasks and often outperforms pre-trained foundation models (Han et al., 2023; Wu et al., 2023; Chen et al., 2023). Nonetheless, training a billion-parameter model often requires significant computational resources and high-quality training datasets to obtain optimal results (Chen et al., 2023). Moreover, this method only offers a temporary solution, as over time the knowledge will be outdated again, leading to a loop of knowledge boundary problem.

Another approach to improve the domain factuality and reduce hallucination of the responses is using the Retrieval-Augmented Generation (RAG) method (Lewis et al., 2021; Izacard and Grave, 2021). The main idea of the RAG method is to use the retriever-reader framework to combine information retrieval (Karpukhin et al., 2020) with LLMs. Specifically, the RAG uses a retriever to select a set of relevant knowledge to the questions, which helps narrow down the answerable evidence for the LLM-based reader. The reader then synthesizes the answer to the query based on the given information like in an "open-book" exam. Therefore, this approach offers the advantage of providing external knowledge to LLMs without requiring the fine-tuning of the models. The responses are also more reliable due to the augmentation of retrieved contexts, which serve as the supporting evidence. Besides, RAG is suitable for both open-domain and closed-domain question-answering tasks, and can also support private use cases depending on the configuration of external data sources.

A recent study showed that the performance of RAG can be hindered by false retrieval, where the system fails to provide accurate information to the generators (Barnett et al., 2024). Several methods aiming to enhance the retriever's performance involve improving query-text embeddings to re-rank pre-retrieved passages (Nogueira and Cho, 2020; Mao et al., 2021; Askari et al., 2023). However, these methods rely solely on textual methods, which may not always be optimal. For example, text embedding models regularly treat input passages independently and do not capture global relations among them (Yu et al., 2022). This inability to capture the dependencies between related passages can potentially restrain the performance of the system (Min et al., 2020). A possible approach to address this problem is to use knowledge graphs (KG) in conjunction with textual information to enhance the retriever in question-answering systems (QAS), providing more robust text representations. This approach has been investigated by studies such as joint representation learning with two modalities to improve re-ranking and answering performance (Yu et al., 2022; Zhou et al., 2020; Ju et al., 2022). Although these methods have shown better results compared to using text input alone, their algorithms and training processes are often costly and complex.

To address the retrieval challenges, this paper aims to enhance the retriever with a comprehensive embedding component that combines both textual data and knowledge graphs (KGs). Unlike previous studies, we propose a simple hybrid pipeline for generating these representations, rather than training a complex joint learning model. The method and experiment were conducted within the use case of a health-tech venture builder, where questions were posed to find answers in proprietary health-related documents. However, the hybrid embedding method has the potential to be domain-agnostic, as long as its components are customized for specific contexts. The experimental results show that our hybrid method helps improve not only the information retrieval performance but also the generative response quality from different LLM-based readers.

2 Related Work

In QAS using the "retrieve-then-read" mechanism, the text embedding plays an important role in the retriever component. State-of-the-art retrievers use dense text embedding methods, often utilizing neural networks like BERT (Devlin et al., 2019) to encode the semantic meaning of the text into dense vectors (Karpukhin et al., 2020; Reimers and Gurevych, 2019; Xiao et al., 2023). Typically, the Dense Passage Retriever (DPR) (Karpukhin et al., 2020) framework learns embeddings for questions and passages using two separate BERT networks with metric learning. Sentence-BERT (Reimers and Gurevych, 2019) also employs metric learning but allows a single BERT to learn embeddings for two sentences through a shared-weight configuration. DPR and Sentence-BERT have also become fundamental approaches to pre-train other general-purpose BERT-based embedding models such as BAAI General Embedding (BGE) (Xiao et al., 2023), General Text Embedding (GTE) (Li et al., 2023b), E5 (Wang et al., 2022), etc. These models can be further fine-tuned for specific downstream tasks(Choi et al., 2021). However, these frameworks were not originally designed to capture the semantic connections between different passages. This drawback can be tackled by taking advantage of structural information captured by knowledge graph solutions. For instance, the Knowledge-aid open-domain QA (KAQA) framework (Zhou et al., 2020) used two KGs representing the relationship between the question and document, and between retrieved documents to support re-ranking the retriever. Also, to improve the retriever by re-ranking, KG-FiD (Yu et al., 2022) used inter-passage relation KG with the graph attention network (Veličković et al., 2018) to update the representation vectors of the passage. Min et al.(Min et al., 2020) introduced an extended passage-level KG and integrated it into the retriever and reader to improve context coverage and response accuracy. In the health domain, (Wise et al., 2020) introduced a KG describing the relationships between scientific articles on COVID-19 and used TransE-based embeddings for article retrieval and recommendation. While these studies obtained impressive results, their implementation and training processes involved the integration of multiple intensive computational models. Inspired by these works, however, the primary goal of this paper is to achieve a more straightforward implementation approach that leverages the knowledge graph to improve conventional retriever and RAG performance.

3 Proposed Method

The conceptual pipeline of the proposed retriever used in the RAG system is illustrated in Figure 1.

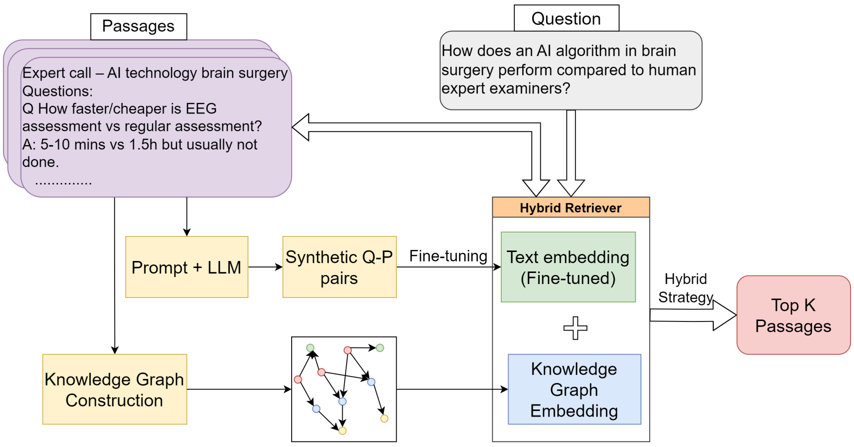

Passages
Expert call – AI technology brain surgery
Questions:
Q How faster/cheaper is EEG assessment vs regular assessment?
A: 5-10 mins vs 1.5h but usually not done.
....
Question
How does an AI algorithm in brain surgery perform compared to human expert examiners?
Hybrid Retriever
Prompt + LLM
Synthetic Q-P pairs
Fine-tuning
Text embedding (Fine-tuned)
Hybrid Strategy
Top K Passages
Knowledge Graph Construction
Knowledge Graph Embedding

Figure 1: The overall conceptual design of our hybrid retriever implemented in the RAG system.

The goal is to improve the quality of the Top-K passages retrieved in response to a given question. This is achieved by measuring the similarity between hybrid embeddings, which combine text and knowledge graph representations of both the Question and the Passages. To adapt the text embedding model to the use case, a fine-tuning process is performed with a synthetic dataset generated by an LLM. Next, an automatic KG construction is proposed, aiming to present the global relationship between Passages. Lastly, we will introduce a strategy aimed at effectively integrating these two modalities into the hybrid retriever.

3.1 Data collection and description

The data for this study consists of interviews with medical experts from a private use case provided by a venture builder, focusing on the technical and business aspects of their medical innovations. The interviews are documented and categorized into 3 main topics, including medical technology, biotechnology, and digital/AI. These documents serve as the knowledge source for the RAG system for the use case.

Firstly, the interview documents stored in the company's database spanning the last 4 years are collected. Document lengths vary, averaging 697 words. After removing the documents with insufficient content, the dataset contains $1,487$ documents. Next, each document undergoes automated pre-processing steps, including the removal of special tokens, images, and tables, as well as English translation.

Finally, the documents are chunked into smaller passages to enhance searchability in the retrieval stage while optimizing computational resources. We choose the chunk size of 512 tokens to fit the small-size BERT-based models in the text embedding step. Furthermore, two adjacent passages of a document are set to overlap by 20 tokens, ensuring a smooth transition of context between them. After the chunking step, the processed dataset comprises a total of 5607 passages.

3.2 Fine-tuning text embedding model

In the context of this paper, the user input questions and the passages' context are distinct, integrating various aspects of the health-tech industry. Hence, using pre-trained text embedding models in model zoos may not sufficiently capture these nuances for retrieval purposes. To address that issue, a BERT-base embedding model is fine-tuned through the training process of the Sentence-BERT (SBERT) framework (Reimers and Gurevych, 2019) (see Appendix A) on our custom dataset.

3.2.1 Constructing training set

The fine-tuning dataset consists of positive query and source passage pairs. Manually annotating these pairs from interview documents is time-consuming, so to simulate real-life scenarios, we use generative LLMs to comprehend the source passages and automatically generate corresponding queries. In each positive pair, the synthetic query is crafted to reflect the questions related to specific information in the source passage, which serves as the ground truth for the query in the retrieval task.

Given that the documents in this study contain private and sensitive information, local and open-source LLMs were selected for query generation instead of using services like OpenAI API. Concretely, small-size but high-performing generative LLMs such as the Zephyr-7B-beta model (Tunstall et al., 2023) and Mistral-7B-OpenOrca [1] are chosen to generate synthetic queries. For each source passage, a single query is synthetically generated by inserting this passage into a prompt to instruct the LLMs. For generalization, we use the Zephyr-7B-beta model for generating the training queries and Mistral-7B-OpenOrca for constructing the test ones. In addition, the instruction prompts for the training and test sets are modified to be slightly different. In this work, these instructions follow a role-playing prompting strategy (Li et al., 2023a). For example, the LLM can be instructed to take on the role of a teacher with the task of generating exam questions based on the passages (see Appendix B). To ensure accurate LLM responses, the prompts were carefully designed, and a set of generated questions was reviewed to confirm they resembled real-world queries.

3.2.2 Training process

The BERT-based model in SBERT is fine-tuned using the Multiple Negative Ranking loss (MNRL) function (Henderson et al., 2017). Mathematically, the loss function is optimized by minimizing the mean negative log-probability of the positive pairs, shown as follows:

$$\begin{aligned} L(\mathbf{q}, \mathbf{p}) &= -\frac{1}{K} \sum_{i=1}^{K} \log \left(P(q_i, p_i)\right) \\ &= -\frac{1}{K} \sum_{i=1}^{K} \log \left(\frac{e^{S(q_i, p_i)}}{\sum_{j=1}^{K} e^{S(q_i, p_j)}} \right) \end{aligned} \tag{1}$$

in which, for a batch size of K, there are K input queries $\mathbf{q} = (q_1, ..., q_K)$ and K corresponding passages $\mathbf{p} = (p_1, ..., p_K)$. The positive pair is denoted as (q_i, p_i) for every $i \leq K$ while the negative pair is indicated as (q_i, p_j) with $i \neq j$ and $i, j \leq K$. To optimize the loss, Adam with decoupled weight decay (AdamW) (Loshchilov and Hutter, 2019) is used along with the warm-up decrease of learning rate enhances for better stability and generalization during training.

In the inference stage, the BERT-based model is taken out of SBERT and used independently. To assess the effectiveness of fine-tuned text embeddings, we then evaluate the retriever's performance on the test set, comparing it to the retrievers using only pre-trained embeddings.

3.3 Knowledge Graph Construction and Embedding

In the proposed system, the role of the KG is to model the connections and relationships between different passages in the dataset. The KG is then embedded in a vector space by a knowledge graph embedding (KGE) model, such that the structural features between passages are preserved through their vector representations.

3.3.1 Knowledge Graph Construction

The KG consists of a set of triples in the form of *(head, relation, tail)* and is constructed following a schema depicted in Figure 2.

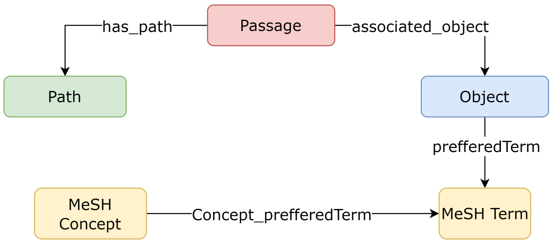


Figure 2: The schema to formalize the knowledge graph construction.

In this schema, the head and tail entities from the triples are categorized into one of five classes, with the Passage, Path, and Object classes serving as general base classes. To enhance the schema for the health-related domain, we incorporate the Medical Subject Headings (MeSH) concepts, MeSH term classes, and their relationships from the MeSH RDF-linked data (Lipscomb, 2000). MeSH, originally a biomedical vocabulary used for indexing and searching PubMed articles, is integrated to enrich the KG with relevant terms. Detailed information about the meaning of entity classes and how these entities are created is described as follows:

Passage entities: The entities belonging to the Passage class represent the text chunks in our dataset. Each Passage entity is defined by its ID in the database.

Path entities: This type of entity represents the path address of the source documents (i.e. the documents before being split into passages) in the database.

Object entities: This type of entity represents the general and bio-medical annotations from the passage. To extract the object entities from the text passages, we leverage the "en_core_sci_lg" model from the SciSpacy library (Neumann et al., 2019) as a Named-entity Recognition (NER) tool. Addi-

---

[1] `https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca`

tionally, each object is linked to a MeSH Descriptor (i.e. a term that describes the main subject of an article) by a Name Entity Linking function. It leads to a total of $65,282$ objects extracted from our passages, in which 6901 objects have linked MeSH Descriptors.

MeSH Concept entities: The MeSH Concept class describes a unit of meaning. In other words, every term in MeSH which is strictly synonymous with each other is grouped into a "Concept". In MeSH, each Descriptor consists of one or more Concepts. Therefore, the MeSH Concept entities in our KG are then retrieved by using a SPARQL[2] query based on the MeSH Descriptor.

MeSH Term entities: The MeSH Term class describe human-readable names used by a MeSH Concept or MeSH Descriptor. A MeSH Descriptor have one MeSH Term while A MeSH Concept can have one or multiple MeSH Terms and they are strictly synonymous. All MeSH Terms are retrieved by SPARQL query based on MeSH Concepts and MeSH Descriptors.

Additionally, entities of different classes are connected by 4 relations. The descriptions of relations used to link head and tail entities are described as follows:

associated_object: This relation describes the connection between Passage and Object entities. It demonstrates what object entities are mentioned in the text.

has_path: This relation connects between Passage and Path entities, indicating the paths where the passages are located.

preferredTerm: This connection between Object and MeSH Term entities indicates which term the Object entity is preferred to refer to.

Concept_preferredTerm: The connection between MeSH Concept and MeSH Term entities, describing the synonym relation.

An example of a subgraph and detailed statistics of the KG is demonstrated in Appendix C.

3.3.2 Knowledge Graph Embedding

The knowledge graph after being constructed is then represented in the vector space by a knowledge graph embedding (KGE) model such that the graph properties are preserved. Although the methodology is applicable to any KGE model, we opt for translational KGE models because of their simplicity and high efficiency.

Generally, translational KGE models operate by using relation embeddings as translations in vector space between head and tail entities. The objective is to learn the embedding of entities and relations in triples to minimize the scoring function $f_r(\mathbf{h}, \mathbf{t})$ of each triple *(h, r, t)* where r is the relation, h and t are head and tail entity embeddings, respectively. Table 1 shows the scoring functions of different KGE models used in this paper.

Table 1: Scoring function equations for TransE, RotatE, and QuatE models. $\circ$ denotes element-wise product, $\otimes$ denotes Hamilton product, $||\cdot||_2^2$ denotes the L2 norm.

Model	Scoring Function
TransE (Bordes et al., 2013)	$\|\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|\|_2$
RotatE (Sun et al., 2019)	$\|\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|\|_2$
QuatE (ZHANG et al., 2019)	$\|\|\mathbf{h} \otimes \mathbf{r} - \mathbf{t}\|\|_2$

The scoring function is optimised through a Margin Ranking loss function, shown as follows:

$$\mathcal{L}(f_p, f_n) = \max(0, f_p - f_n + \lambda) \tag{2}$$

in which λ denotes the margin value, f_p and f_n are the scoring function values of a positive triple (i.e. the actual triple in KG) and a negative triple (i.e. the non-existent triple in KG), respectively. This loss function aims to encourage the model to improve its embedding representations and effectively distinguish between positive and negative triples. To sample negative triples, the head or tail of a positive triple is randomly swapped with an entity from another one in the training batch. This process is carefully engineered to ensure that the resulting corrupted triples do not already exist as positive examples in the original KG. Finally, the objective of learning embedding for entities and relationships can be achieved by using the Stochastic Gradient Descent algorithm.

3.4 Hybrid Retrieval Strategy

Figure 3 illustrates the process of our hybrid retrieval strategy. The strategy is divided into five main steps as follows:

Step 1: Given a question, a set of top N (N > K) relevant passages are retrieved using the cosine similarity of their text embeddings. This step aims to narrow the search space by filtering out irrelevant passages based on their semantic nuances.

Step 2: From the top N retrieved passages, their text embeddings and KG embeddings are horizontally concatenated. In the concatenation vectors,

---

[2] `https://www.w3.org/TR/sparql11-query/`

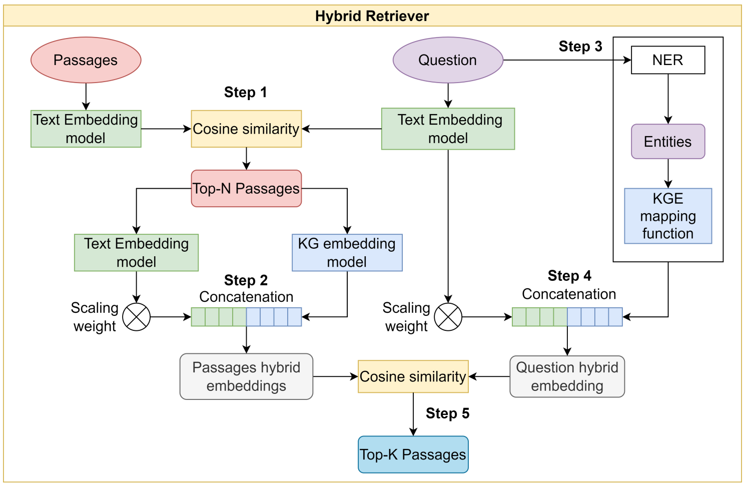


Figure 3: The hybrid retrieval strategy aims to utilise the fusion of text and KG embeddings.

we prioritize the impact of the text embedding component to emphasize the importance of semantic nuances, while utilising the KG embedding component as an auxiliary extension. Thus, text embeddings are multiplied element-wise by a scaling weight α (where $\alpha > 1$) to amplify their significance. The concatenation operation is then specifically formulated as follows:

$$Concat(TE, KGE) = [TE * \alpha, KGE] \quad (3)$$

where TE and KGE indicate a text embedding and a knowledge graph embedding respectively. With the amplification of text embedding, the hybrid representations of passages have more internal semantic features while still containing the global relationship information captured by KG-based vectors.

Step 3: Since the question is not explicitly modelled in our KG, approximating its representation in the KG vector space is needed. Accordingly, all objects in the question are first extracted by using the same NER model in Section 3.3. Next, the question embedding is then approximated by a mapping function, shown as follows:

$$\mathbf{q} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{p}_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \text{passage\_has\_obj}(p_i, o_j)} \quad (4)$$

where M is the number of objects extracted in the question, N is the number of top N retrieved passages, $\mathbf{q}$ is the approximated embedding of a question, $\mathbf{p}_{ij}$ is the embedding of passage p_i in top N that contains object o_i, and

$$\text{passage\_has\_obj}(P_i, O_j) = \begin{cases} 1 & \text{if } p_i \ni o_j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Step 4: The text embedding and the approximated KGE of the question are concatenated to form the hybrid vector. Also, the same weight α value is applied to the text embedding of the question before concatenation.

Step 5: Finally, the top K passages are retrieved from the N passages by taking the K highest cosine similarity scores between the hybrid vectors of the question and the N passages.

4 Experiments

4.1 Experimental Setup

4.1.1 Text embedding model setup

For the text embedding component in the retriever, three small-size BERT-based models were selected as the baselines: BGE model (Xiao et al., 2023), E5 model (Wang et al., 2022) and GTE model (Li et al., 2023b). Each model comprises 12 transformer blocks, 12 attention heads per self-attention layer, an input size of 512 tokens, and an embedding size of 384.

The training set to fine-tune the models was generated from 500 random passages, following the method in Section 3.2. The models were trained in 50 epochs, with a batch size of 10 and using the AdamW optimizer. The initial learning rate was configured at 0.001 and with a decay through a warm-up step of 0.1. We trained the models on Google Colab Pro with 15GB VRAM NVIDIA T4.

4.1.2 Knowledge graph embedding configuration

Three KGE models were experimented with, including TransE, RotatE, and QuatE. Each model's embedding size of 200 was configured for optimal performance based on grid search. The models were trained on 20 epochs. Each training batch consisted of 128 positive triples, where for each positive triple, one negative triple was generated.

4.1.3 Hybrid Retriever configuration

The top-N value was initialized at 20. Experimentation included top-K values of 3 and 5. Various combinations of BGE text embeddings and KGE models were tested for the hybrid embeddings. Through grid search, a weight of $\alpha = 2.5$ was found to optimize performance, ensuring the hybrid retriever achieved its peak effectiveness.

4.2 Evaluation Scheme

4.2.1 Passage Retrieval Evaluation

To evaluate the retriever, three synthetic test sets were created, denoted as the first, second, and third test sets, following the guidelines outlined in Section 3.2. The queries from the first and second test sets were generated from the same random 200 passages. In the first test set, one question was synthesised per passage, while in the second test set, the ratio was 2:1. The third test set was created from a different set of 200 passages, with a 1 question per passage ratio.

The retriever performance was measured by two metrics: Hit Rate (HR) for retrieval accuracy and Mean Reciprocal Rank (MRR) for ranking ability. The higher the metric values, the better the performance.

4.2.2 RAG performance evaluation

To evaluate an end-to-end RAG, another test set was used that included 50 golden question-answer pairs manually extracted from the documents. Given the question, the correctness of RAG's response was measured by comparing it to the golden answer, using Semantic Answer Similarity (SAS) score (Risch et al., 2021). SAS score is calculated by a cross-encoder model given the predicted answer and golden answer pairs. In this paper, the cross-encoder version of the BGE model was used. The SAS score also measured the level of relevancy between the RAG responses and the retrieved passages.

4.3 Experimental Results

4.3.1 Passage Retrieval Results

The comparison of retrievers using different embedding models is demonstrated in Table 2. Across all test sets, retrievers using the fine-tuned text embedding models exhibited superior performance compared to their pre-trained counterparts in both HitRate@K and MRR@K metrics. Notably, the fine-tuning of GTE and E5 models resulted in more pronounced enhancements. Although the fine-tuned BGE model showed only a slight improvement over its pre-trained version, both iterations of BGE outperformed the E5 and GTE models. These improvements were consistent across all top-K scenarios. These findings also suggest that relying solely on general-purpose text embedding models, which leverage pre-trained knowledge, may not yield optimal results for domain-specific datasets. The proposed fine-tuning solution can significantly enhance performance and stabilize results in such cases.

On the other hand, our method of integrating fine-tuned BGE with any knowledge graph embedding model demonstrated notable enhancements compared to BGE-only retrievers across all test sets and Top-K settings. Particularly, the hybrid approach showed increases of up to 8.1% in HitRate and up to 8.7% in MRR. Notably, leveraging TransE embeddings, the hybrid retriever produced the highest results compared to other hybrid variations. This emphasizes the effectiveness of leveraging global semantic relationships to enrich the textual representation of both passages and queries, hence, enhancing the overall retrieval performance.

4.3.2 End-to-end RAG performance

The effectiveness of the hybrid retriever was further evaluated through its impact on the answering performance of RAG systems. For this experiment, the hybrid retriever employed fine-tuned BGE and TransE models and returned the top 3 passages. The performance of LLM-based readers was then compared across three conversational LLMs: LLaMA-2-13B-chat (Touvron et al., 2023), Zephyr-7B-beta and Mistral-7B-OpenOrca.

Table 3 displays the response quality scores for various RAG settings. In terms of Relevancy, the scores remained relatively consistent, suggesting that all LLM baselines could properly answer questions following the retrieved contexts. However, our analysis of Correctness scores revealed a notable enhancement in RAG performance when utilizing hybrid retrievers, with improvements of up to 13.1%. This highlights the significant impact of hybrid retrievers on the accuracy of RAG-generated responses.

Table 2: The performance comparison between retrievers using hybrid embeddings and those using only text embeddings.

Retriver	1st Test set		2nd Test set		3rd Test set	
	HR	MRR	HR	MRR	HR	MRR
Top K = 3						
GTE$_{\text{pre-trained}}$	0.3865	0.2938	0.3291	0.2324	0.4690	0.3427
E5$_{\text{pre-trained}}$	0.6546	0.5567	0.5753	0.4882	0.6804	0.5506
GTE$_{\text{fine-tuned}}$	0.7989	0.6993	0.6934	0.5644	0.7886	0.6683
E5$_{\text{fine-tuned}}$	0.7525	0.6125	0.6231	0.5201	0.7474	0.6091
BGE$_{\text{pre-trained}}$	0.8247	0.6941	0.7311	0.6139	0.8195	0.6821
BGE$_{\text{fine-tunned}}$	0.8350	0.7164	0.7437	0.6335	0.8350	0.6941
BGE$_{\text{ft}}$ + TransE	**0.8917**	**0.7506**	**0.8040**	**0.6892**	0.8814	0.7336
BGE$_{\text{ft}}$ + RotatE	0.8763	0.7526	0.7839	0.6570	**0.8865**	**0.7431**
BGE$_{\text{ft}}$ + QuatE	0.8711	0.7465	0.7814	0.6440	0.8763	0.7250
Top K =5						
GTE$_{\text{pre-trained}}$	0.4742	0.3136	0.3994	0.2487	0.5515	0.3610
E5$_{\text{pre-trained}}$	0.7577	0.5798	0.6482	0.5051	0.7989	0.5774
GTE$_{\text{fine-tuned}}$	0.8608	0.7129	0.7587	0.5795	0.8556	0.6838
E5$_{\text{fine-tuned}}$	0.7938	0.6223	0.6909	0.5356	0.7886	0.6189
BGE$_{\text{pre-trained}}$	0.8505	0.6993	0.7989	0.6292	0.8659	0.6926
BGE$_{\text{fine-tunned}}$	0.8917	0.7298	0.8090	0.6485	0.8917	0.7080
BGE$_{\text{ft}}$ + TransE	**0.9329**	**0.7652**	**0.8542**	**0.7007**	**0.9381**	**0.7435**
BGE$_{\text{ft}}$ + RotatE	0.9175	0.7616	0.8517	0.6726	0.9175	0.7498
BGE$_{\text{ft}}$ + QuatE	0.9072	0.7542	0.8316	0.6560	0.9175	0.7341

Table 3: Responses comparison between different RAG's combinations.

LLM reader	Retriever	Relevancy	Correctness
LLama-2-13B-chat	BGE ft	0.9846	0.7655
Mistral-7B-OpenOrca	BGE ft	0.9909	0.7193
Zephyr-7B-beta	BGE ft	0.9708	0.7486
LLama-2-13B-chat	hybrid	0.9921	0.7729
Mistral-7B-OpenOrca	hybrid	0.9891	0.8136
Zephyr-7B-beta	**hybrid**	**0.9878**	**0.8283**

The experimental results also correlated with the correctness of answer examples shown in Table 7 (see Appendix F). The answers from the Zephyr-7B-beta model based on hybrid retriever contexts were more detailed and aligned better with the golden answer than those from BGE-only retriever contexts, which contained less information and thus, had a lower correctness score.

5 Discussion

In this section, we will discuss how the hybrid embeddings help to improve the retriever by analyzing the impacts of the weight α. Additionally, a comparison of the proposed method with other re-ranking mechanisms will be demonstrated. In these analyses, we experimented with a sub-case using our 1st retriever test set introduced in Section 4.1.

As shown in Figure 6 (see Appendix D), it is clear that the value of α greatly affects the performance of the hybrid retriever. When α was set to 0, it was equivalent to the case of a retriever only using TransE embeddings. However, the performance of the retriever in that case was poorly underperformed, especially when the top-N value increased. This observation suggests that when the retriever relies solely on KGE and retrieves information from a larger pool, it is more prone to noise and irrelevant information. When α was set to 1, text and KG embeddings were equally concatenated. While this improved hybrid retriever performance, it decreased at higher top-N values. This finding indicates that balanced weights might cause KG embeddings to diminish the semantic meaning of text embeddings in their hybrid vectors, resulting in unstable outcomes.

Conversely, as the parameter α increases, it boosts the influence of text embeddings. This, in turn, strengthened the semantic features in the combined vectors, resulting in better performance for the retriever. Notably, with larger α values, the performances remained relatively unchanged by variations in top-N values. This observation indicates that the α factor can aid hybrid vectors in differentiating dissimilar ones. However, it is crucial to keep α values within an appropriate range. If α is too large, the text embedding features can overshadow the global features from KGE, causing the hybrid vectors to resemble text embeddings too closely.

Additionally, our two-stage hybrid retrieval strategy can be considered to be similar to the "retrieve then re-rank" mechanism. However, instead of using a re-ranking model, the hybrid vectors take the KGE component to re-rank the pre-defined passage orders. The results show that our hybrid retriever had comparative performance compared to other "retrieve then rerank" paradigms despite not being intentionally designed for re-ranking purposes (see Appendix E). Furthermore, the proposed hybrid retriever only uses cosine similarity to retrieve passages, which is computationally lighter for inference than using neural-based re-ranker models.

6 Conclusion

In this work, a hybrid method is introduced which leverages both text and knowledge graph embeddings to advance the retriever in Retrieval-Augmented Generation systems. The text embedding model is fine-tuned with a synthetic dataset to adapt to downstream tasks. Meanwhile, the KGE component is trained from a KG presenting the global relationships between passages in the dataset. Additionally, the proposed hybrid retrieval strategy efficiently integrates these two representation types without using complex architecture or training processes typical in other KG-based retrieval methods.

The experimental results demonstrate that the method can significantly improve the retriever's performance in both accuracy and ranking in comparison to the baseline methods. This improvement subsequently results in the higher correctness of the RAG's responses.

In this paper, the methods were tested in an application in the health-tech domain where knowledge can be represented based on Medical Subject Headings (MeSH) classes. However, the proposed method is generic and can be applied to any domain where the KG can be meaningfully constructed to describe the relations between different passages.

References

Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 Score as Text Improves BERT-Based Re-rankers. ArXiv:2301.09728 [cs].

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven Failure Points When Engineering a Retrieval Augmented Generation System. ArXiv:2401.05856 [cs].

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. ArXiv:2311.16079 [cs].

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks. ArXiv:2101.10642 [cs].

Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. 2023. ChatGPT and the rise of large language models: the new AI-driven infodemic threat in public health. *Frontiers in Public Health*, 11:1166120.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K. Bressem. 2023. MedAlpaca – An Open-Source Collection of Medical Conversational AI Models and Training Data. ArXiv:2304.08247 [cs].

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. ArXiv:1705.00652 [cs].

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ArXiv:2311.05232 [cs].

Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. ArXiv:2007.01282 [cs].

Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022. Grape: Knowledge Graph Enhanced Passage Reader for Open-domain Question Answering. ArXiv:2210.02933 [cs].

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. ArXiv:2004.04906 [cs].

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. ArXiv:2005.11401 [cs] version: 4.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. ArXiv:2303.17760 [cs].

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards General Text Embeddings with Multi-stage Contrastive Learning. ArXiv:2308.03281 [cs].

Carolyn E. Lipscomb. 2000. Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–266.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. ArXiv:1711.05101 [cs, math].

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-Guided Passage Reranking for Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 344–350, Online. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering. ArXiv:1911.03868 [cs].

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327. ArXiv:1902.07669 [cs].

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. ArXiv:1901.04085 [cs].

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv:1908.10084 [cs].

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic Answer Similarity for Evaluating Question Answering Models. ArXiv:2108.06130 [cs].

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? ArXiv:2002.08910 [cs, stat].

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. ArXiv:1503.03832 [cs].

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. ArXiv:1902.10197 [cs, stat].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv:2302.13971 [cs].

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct Distillation of LM Alignment. ArXiv:2310.16944 [cs].

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. ArXiv:2212.03533 [cs].

Colby Wise, Vassilis N. Ioannidis, Miguel Romero Calvo, Xiang Song, George Price, Ninad Kulkarni, Ryan Brand, Parminder Bhatia, and George Karypis. 2020. COVID-19 Knowledge Graph: Accelerating Information Retrieval and Discovery for Scientific Literature. ArXiv:2007.12731 [cs].

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. PMC-LLaMA: Towards Building Open-source Language Models for Medicine. ArXiv:2304.14454 [cs].

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. ArXiv:2309.07597 [cs].

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering. ArXiv:2110.04330 [cs].

SHUAI ZHANG, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mantong Zhou, Zhouxing Shi, Minlie Huang, and Xiaoyan Zhu. 2020. Knowledge-Aided Open-Domain Question Answering. ArXiv:2006.05244 [cs].

A Sentence-BERT architecture

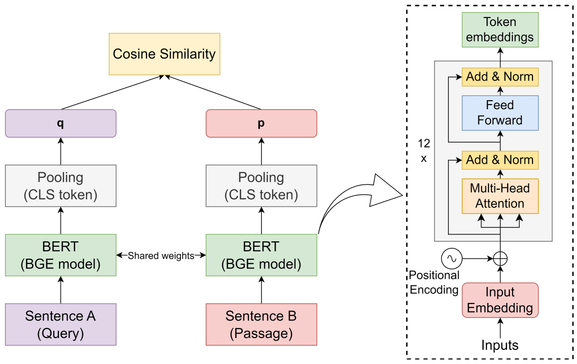


Figure 4: A structure of Sentence-BERT using a Siamese network. Query (Sentence A) and Passage (Sentence B) are encoded separately by two shared-weight BERT-base blocks.

As shown in Figure 4, SBERT uses a Siamese network (Schroff et al., 2015) with two shared-weight BERT-based models. Given the query and passage pairs, SBERT will learn to give higher cosine similarity scores for matched pairs and vice versa. In principle, the embedding model used in SBERT can be any variation of BERT.

In this paper, all layers of the BERT-based model are fine-tuned to ensure complete adaptation to the training set. The *Pooling* layers in SBERT are configured depending on pre-trained BERT to aggregate contextualized word embeddings of query and passage resulting vectors q and p, respectively. Finally, the cosine similarity is then computed between them.

B Prompts for synthetic question generation

The instructional prompt templates used for the Zephyr 7B and Mistral 7B models for generating train and test sets in our case are depicted in Table 4. In both prompts, a role-playing prompting strategy (Li et al., 2023a) was chosen to guide the models' behaviours and responses.

C Graph statistics

An example of a subgraph is shown in Figure 5. Table 5 shows the number of entities in each class and the relations in our KG. It is built by $455,737$ triples including $98,524$ entities, in which there are $5,607$ Passage entities. In $65,282$ objects, each Object can be associated by an average of 5.6 Passages and by at least 1 Passage. Besides, there are $23,634$ Objects linked with MeSH Terms. On average, each Object is connected to 2.7 MeSH Terms while each MeSH Term is referred to by 3.26 Objects. Furthermore, each MeSH Concept is referred to by an average of 2.1 MeSH Term. For the Path entity, on average, each Path is connected by 3.77 Passages, in which the highest number of Passage originating from a Path is 20.

Table 4: Two prompt templates to generate synthetic training and testing sets. The prompt for the train set is inputted to the Zephyr-7B-beta model while the prompt for the test set is used to guide Mistral-7B-OpenOrca.

Prompt for train set
Context information is below. --------------------- {context_input} --------------------- Given the context information and no prior knowledge, you are a Teacher/Professor. Your task is to set up {num_questions_per_chunk} questions for an upcoming quiz/examination. The question must be based on the main context. Additionally, the question must have a clear answer indicated in the context information. Finally, return the question with a question mark at the end.
Prompt for test set
Context information is below. --------------------- {context_input} --------------------- Given the context information and no prior knowledge, you are a Teacher/Professor. Your task is to set up {num_questions_per_chunk} questions for an upcoming quiz/examination. The questions should be diverse in nature across the document. Restrict the questions to the context information provided.

Table 5: Summary of knowledge graph details

Entity Class	**Count**	**Relation Type**	**N.o triples using relation**
Passage	5607	associated_object	370936
Path	1487	has_path	5607
Object	65282	preferredTerm	64053
MeSH Concept	6901	Concept_prefferedTerm	15141
MeSH Term	19247		
Total	98524	Total	455737

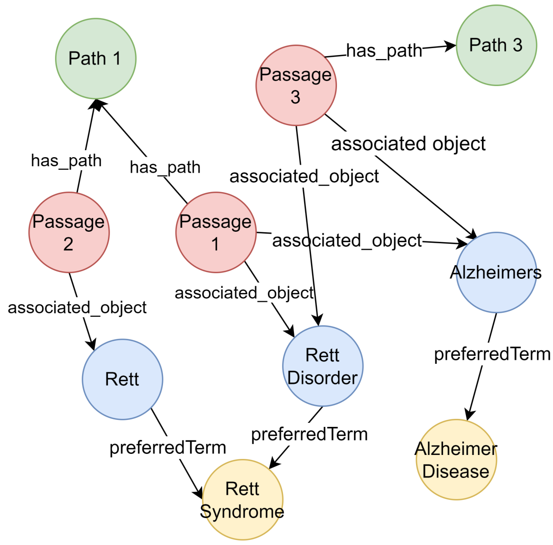
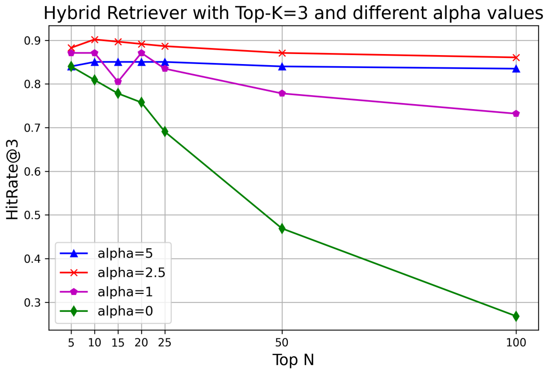


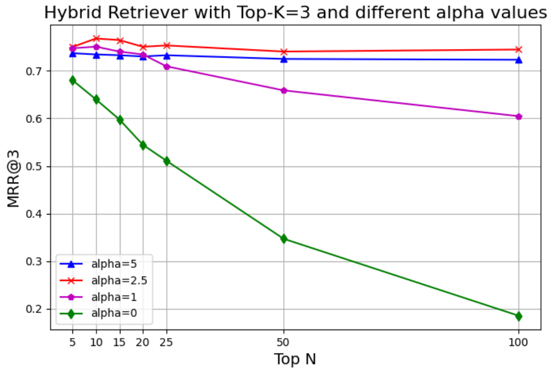
Figure 5: An example of a sub-graph illustrates the relationship between different passages. Intuitively, Passages with more common connections are located closer in the graph.

D Influence of scaling weight α

Figure 6 presents the results of the hybrid retriever's performance employing fine-tuned BGE and TransE embeddings, concerning HitRate@3 across varying α values.



(a) Performance Analysis with HitRate@3



(b) Performance Analysis with MRR@3

Figure 6: Overall Performance Analysis on HitRate@3 and MRR@3 with different alpha values

E Comparision to "Retrieve then Re-rank" mechanism

Table 6 shows the comparison between the proposed methodology and retrievers integrating reranking models.

Table 6: The performance comparison on the first test set between our hybrid retriever and different combinations of retriever and re-ranker models.

Retriever + Reranker	HR@3	MRR@3
BGE$_{\text{ft}}$	0.8350	0.7164
BGE$_{\text{ft}}$ + Stsb-roberta-base	0.3762	0.2689
BGE$_{\text{ft}}$ + ms-marco-miniLM-L-2-v2	0.8195	0.6701
BGE$_{\text{ft}}$ + BGE-rerank-base	0.9123	0.7439
BGE$_{\text{ft}}$ + ms-marco-miniLM-L-6-v2	**0.9175**	**0.8092**
BGE$_{\text{ft}}$ + TransE (ours)	0.8917	0.7506

F Examples of system's response

Table 7 illustrates an example of answers from the Zephyr-7B-beta model for a question based on the contexts provided by BGE-only and hybrid retrievers. The answer generated from the hybrid retriever's contexts exhibited greater detail and closer alignment with the golden answer. In contrast, the answer obtained from the BGE retriever's passages contained only half of the information leading to a lower correctness score.

Table 7: Comparison of answers from Zephyr-7B-beta reader with different retrievers

Question: What is the main reason for revision surgery?
Golden Answer: Infection, infection and infection. We already have antibacterial coatings. However, these do not cover for the 100%. Furthermore, the proximal and distal parts are coated, but the valve is not.
BGE + Zephyr-7B-beta: According to the expert call, the main reason for revision surgery in brain shunts is infection, infection, and infection. **Correctness score:** 0.4096
Hybrid + Zephyr-7B-beta: The main reason for revision surgery in the context provided is infection, infection, and infection. The expert mentions that if there is no native increased pressure, then the shunts get clogged due to pathology from the past such as high protein levels, haemorrhages, or circulating tumour cells. The valve is prone to erythrocyte cells, and the main reason for revision surgery is infection. The expert also mentions that they already have antibacterial coatings, but these do not cover 100% and the proximal and distal parts are coated, but the valve is not. Therefore, infection remains a significant issue in shunt revision surgeries. **Correctness score:** 0.9975

Investigating Gender Bias in Large Language Models Through Text Generation

Shweta Soundararajan and **Sarah Jane Delany**
Technological University Dublin
`shweta.x.soundararajan@mytudublin.ie`; `sarahjane.delany@tudublin.ie`

Abstract

Large Language Models (LLMs) have swiftly become essential tools across diverse text generation applications. However, LLMs also raise significant ethical and societal concerns, particularly regarding potential gender biases in the text they produce. This study investigates the presence of gender bias in four LLMs: ChatGPT 3.5, ChatGPT 4, Llama 2 7B, and Llama 2 13B. By generating a gendered language dataset using these LLMs, focusing on sentences about men and women, we analyze the extent of gender bias in their outputs. Our evaluation is two-fold: we use the generated dataset to train a gender stereotype detection task and measure gender bias in the classifier, and we perform a comprehensive analysis of the LLM-generated text at both the sentence and word levels. Gender bias evaluations in classification tasks and lexical content reveal that all the LLMs demonstrate significant gender bias. ChatGPT 4 and Llama 2 13B exhibit the least gender bias, with weak associations between gendered adjectives used and the gender of the person described in the sentence. In contrast, ChatGPT 3.5 and Llama 2 7B exhibit the most gender bias, showing strong associations between the gendered adjectives used and the gender of the person described in the sentence.

1 Introduction

Large Language Models (LLMs) have rapidly emerged as indispensable tools in today's digital landscape, revolutionizing text generation across various applications. Their applications span various domains, including the medical domain for tasks like medical report generation, offering recommendations for diagnosis and treatment (Wang et al., 2023b), and generating clinical text data (Dai et al., 2023; Tang et al., 2023). They have been used for generating reference letters (Wan et al., 2023), aiding academic research writing (Sallam, 2023; Transformer et al., 2022), creating children's education materials (Valentini et al., 2023), serving as personal writing assistants (Hallo-Carrasco et al., 2023), and composing item descriptions for recommendation systems (Acharya et al., 2023). Additionally, they have been used to generate data for training data augmentation in low-resource scenarios (Dai et al., 2023; Ubani et al., 2023), fine-tuning multilingual models (Michail et al., 2023), translation (Zhang et al., 2023; Wang et al., 2023a), and quality estimation (Huang et al., 2023) in machine translation.

However, alongside their impressive capabilities, LLMs have also raised significant ethical and social concerns, particularly regarding gender bias in the text they generate. Recent studies have shown that LLM-generated text can contribute to societal harm, notably through the perpetuation of gender bias (Wan et al., 2023; Kotek et al., 2023; Dong et al., 2024; Fang et al., 2024; Ovalle et al., 2023). Gender stereotypes and bias can have a negative impact on minority groups in society. It has been shown, for example, that the use of LLM-generated text containing gender stereotypes in children's stories can influence young minds (Arthur et al., 2008; Bender et al., 2021). Kotek et al. (2023) assert that, according to psychological developmental literature, children internalize societal expectations from a very young age potentially altering their hobbies, interests, and even academic and career paths accordingly. Another consequence of using LLM-generated text becomes evident when LLMs are used to generate recommendation letters, reference letters (Wan et al., 2023), resumes (Zinjad et al., 2024), and job postings[1]. Gender bias in these LLM-generated text can deter women from applying for the position and sabotage application success rates for female applicants (Madera et al., 2009; Khan et al., 2023; Gaucher et al., 2011; Tang et al., 2017).

In this paper, we compare and assess the gender

---

[1] https://northreach.io/blog/

bias in four LLMs namely ChatGPT 3.5, ChatGPT 4, Llama 2 7B, and Llama 2 13B. We generate gendered language[2] datasets by prompting the LLMs to generate sentences about men and women using gender lexicon (gender-coded) words included in the instruction prompt. The masculine-coded and feminine-coded words in a gender lexicon are associated with gender stereotypes and often referred to as gendered wording (Gaucher et al., 2011). Recognizing that adjectives can reflect stereotypical characteristics or traits of a specific gender (Arvidsson, 2009; Fast et al., 2016; Hoffman and Tchir, 1990; Morelius, 2018; Maass, 1999; Ellemers, 2018), we focus on using adjectives. With these datasets, we assess gender bias in the generated text in two different ways - (1) we measure gender bias in a downstream classification task–using the generated data to train a gender stereotype detection task, which involves predicting whether a sentence is consistent with, or contradictory to gender stereotype and measure gender bias in the classifier and (2) performing a data analysis of the generated text at sentence and word level. At a sentence level, we assess the likelihood of LLMs adding additional gendered adjectives (other than those explicitly included in the prompt) in the generated sentences that match the gender of the person described in the sentence. LLMs which are less likely to use additional adjectives that match the gender of the person in the sentence can be considered to be less aligned with gender stereotypes. At a word level, we identify each LLM's assumed gender of adjectives based on the likelihood of the LLM to use specific adjective with certain genders. We then see how these compare with the gender labels given to these adjectives in a gender lexicon. LLMs with fewer matches can be considered to be less biased to gender stereotypes.

Our study reveals that the datasets generated by all LLMs exhibit gender bias in detecting gender stereotypes, as indicated by the results of the downstream classification task with Llama 2 13B showing the least gender bias, while Llama 2 7B demonstrates the highest bias among the LLMs tested. Furthermore, our data analysis at sentence level finds that all LLM are more likely to add additional gendered adjectives that match the gender of the person described in the sentence with ChatGPT 4 showing the weakest association between the gender of the adjectives used and the gender of the person, and ChatGPT 3.5 showing the strongest association. Our analysis of adjective usage by the LLMs finds ChatGPT 4 uses a slightly smaller percentage of adjectives that are gender coded with the gender of the person described in the sentence, than the other LLMs.

Our conclusion is that ChatGPT 4 and Llama 2 13B demonstrate the least gender bias, while ChatGPT 3.5 and Llama 2 7b, the most.

2 Related Work

Several studies have assessed biases in language models. Zayed et al. (2024) and Li et al. (2023) classified bias measurement approaches as intrinsic or extrinsic while Chu et al. (2024) categorized them as embedding-based, probability-based, and generation-based, with the first two falling under intrinsic and the latter under extrinsic. Intrinsic approaches evaluate the bias of the model independently of any downstream tasks. For instance, some works (Caliskan et al., 2017; Wan et al., 2023; May et al., 2019) evaluated bias by statistically quantifying associations between targets and concepts in the embedding space. Other studies have measured bias by analyzing probabilities assigned by LLMs to different options, such as predicting candidate words based on templates (Webster et al., 2020; Kurita et al., 2019), or candidate sentences based on author-created or crowdsourced evaluation datasets (Nadeem et al., 2020; Nangia et al., 2020; Felkner et al., 2023).

Extrinsic approaches assess models' bias within the context of a downstream task and the model generated texts. Benchmark datasets have been used to measure bias in coreference resolution, where models must identify the correct pronoun for a person described by their occupation (Zhao et al., 2018; Rudinger et al., 2018; Levy et al., 2021; Kotek et al., 2023; Ovalle et al., 2023). Gender bias is indicated if the model outputs a pronoun stereotypically associated with that occupation. Question answering tasks have also been used to assess gender bias, where the LLM is is asked to agree or disagree with statements (Morales et al., 2023; Feng et al., 2023), or to answer multiple-choice questions (Parrish et al., 2021). Summarization tasks assess gender bias by coding the presence or absence of specific information in the LLM-generated summaries (Acerbi and Stubbersfield, 2023). Classification tasks have also been used, using an auxiliary

---

[2] Gendered language refers to the use of words that indicate the gender of an individual.

model to assess gender bias in the generated text. If the auxiliary model classifies text generated using similar prompts but featuring distinct social groups differently, then the generated text is biased (Chu et al., 2024). For example, De-Arteaga et al. (2019) measured gender bias in occupation classification using the Bias-in-Bios dataset by examining the difference in true positive rates between genders. Wan et al. (2023) generated reference letters for men and women using LLMs, classified them as agentic or communal, and measured gender bias using a statistical t-test. Other studies (Morales et al., 2023; Dhamala et al., 2021) assessed bias by prompting the LLM with sentences related to different groups and evaluating the social bias, sentiment, and toxicity of its generated continuations. Chu et al. (2024)'s generation-based approaches for measuring gender bias in LLM generated text also include metrics that look at the distribution of tokens related to one gender group with that of another or similar nearby groups. The most commonly used measure here is the odds ratio, which measures biases in word choices between wordings in documents related to different genders (Sun and Peng, 2021; Wan et al., 2023; Cryan et al., 2020).

3 Approach

We generate examples of gendered language using LLMs based on gender lexicons that contain gender-coded words, i.e. words that are associated with masculine and feminine stereotypes. Figure 1 illustrates this process.

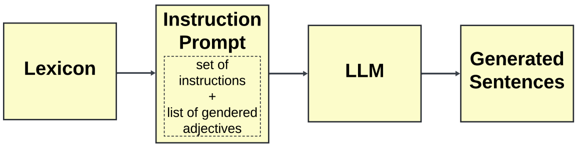


Figure 1: Pipeline for data generation using LLMs

3.1 Lexicon choices

Since adjectives often reflect stereotypical characteristics—such as personality traits and physical appearance—we focus on generating sentences about men and women using adjectives from these lexicons. The two gender lexicons available for use were developed by Gaucher et al. (2011) and Cryan et al. (2020). Gaucher et al.'s (2011) lexicon is a manually curated lexicon that contains 42 masculine words and 40 feminine words. Although not all words in this lexicon are adjectives, they have been stemmed so that they can be used flexibly as adjectives, verbs, or nouns. This lexicon has been widely used to examine gendered wording in job ads (Gaucher et al., 2011). Cryan et al.'s (2020) lexicon is a more recent gender lexicon focused on capturing people's perceptions of gender stereotypes in contemporary society. It contains adjectives labeled with gender scores, where adjectives are identified as masculine or feminine based on how they are perceived by people. To create this lexicon, Cryan et al. (2020) extracted candidate adjectives from Wikipedia and used crowdsourcing to label the most commonly used adjectives with gender scores. These labels were then used to train a model to predict gender scores for other words. However, the lexicon provided by the authors included some non-adjectives and appeared incomplete, missing certain commonly used words evident in the original paper. To address this, Soundararajan et al. (2023) performed pre-processing to remove non-adjectives, resulting in a revised lexicon, which we refer to as *Cryan 1*. This version contains 1845 masculine and 1675 feminine adjectives. Additionally, the most frequently used adjectives, 299 in total, visible in the word cloud diagram in the original Cryan et al. (2020) paper were identified. This subset, consisting of 152 masculine and 147 feminine adjectives, is referred to as *Cryan 2*. These adjectives were used in the prompts to generate examples of gendered language using LLMs.

3.2 Prompt engineering and Prompt design

Prompting (Brown et al., 2020; Chen et al., 2023) drives pre-trained LLMs with task-specific instructions to produce outputs for the task without the need for fine-tuning the LLM. We opted for zero-shot prompting (Brown et al., 2020) for a number of reasons. It can accommodate a broader range of prompts without needing specific examples and our goal was to prevent the LLM from generating sentences solely based on the examples or templates provided in the prompt. Furthermore, we aimed to mitigate the risk of introducing bias by not including examples in the prompt.

We started with the instruction prompt, a baseline, found in Soundararajan et al.'s (2023) work and performed prompt engineering, an iterative process of designing and refining the input instruction prompt to obtain the desired response from the LLMs. We constructed several instruction prompts and selected the final prompt by reviewing the generated sentences. Our prompt refinements involved tweaking this baseline prompt for better clarity, adding and removing instructions, and varying the number of gendered adjectives included in the prompt. Ultimately, we selected a final prompt that included 10 gendered adjectives. This choice was driven by our observation that when many adjectives are included in the prompt, a few LLMs tend to forget or lose information from the prompt while generating output.

> Read the instructions one by one and produce the output once all the instructions have been read.
> 1. Generate 10 sentences about a male/males/man/men/boy/boys where each sentence must strictly use each WORD listed below as an adjective to depict the characteristic or traits of the male/males/man/men/boy/boys being discussed in the sentence.
> WORDS: outspoken, independent, ambitious, assertive, muscular, courageous, handsome, decisive, daring, authoritative
> 2. Use pronouns, nouns, or names of individuals to reference the male/males/man/men/boy/boys discussed in the sentences.
> 3. Include a mix of sentences mimicking styles found in various contexts, such as newspapers, magazines, children's books, job advertisements, storybooks, movie dialogues, and recommendation letters.
> 4. Include a mix of all types of sentences (simple sentences, compound sentences, complex sentences, and compound-complex sentences) in the output.
> 5. Utilize any tenses and parts of speech in the sentences.
> 6. Format the output as follows:
> WORD : type of sentence : sentence

Figure 2: Instruction prompt to generate sentences about men. LLMs were prompted to generate sentences about men and women using both masculine and feminine adjectives.

3.3 LLM choices

We selected seven open-source and closed-source LLMs. These LLMs vary from low to mid-range in terms of parameters: ChatGPT 3.5, ChatGPT 4, Llama 2 7B (7 billion), Llama 2 13B (13 billion), Mistral 7B (7 billion), Falcon 7B (7 billion), and Falcon 40B (40 billion). The parameter counts for ChatGPT 3.5 and ChatGPT 4 are not disclosed as they are closed-source LLMs. Temperature, a hyperparameter in text generation, modulates the randomness or creativity of the LLM's responses. Given our focus on generating gendered language, we are cautious about setting the temperature too high to prevent the LLM from excessively creative outputs or including too many words, which may introduce bias. To ensure consistency we opt for a uniform temperature value across all LLMs, set at 0.75. For all the LLMs except ChatGPT models, we configured the maximum number of tokens to be 1024. All other hyperparameters were left at their default values for each respective LLM.

We provided our prompts to the LLMs and manually reviewed the generated outputs. The outputs are solely based on the version of the LLM available in March 2024, when the LLMs were accessed. ChatGPT 3.5 (gpt-3.5-turbo-0125) and ChatGPT 4 (gpt-4-turbo-preview which points to gpt-4-0125-preview) produced relevant and reasonable outputs consistently. However, the outputs generated by Llama 2 7B (llama-2-7b-chat-hf) and Llama 2 13B (llama-2-13b-chat-hf) models were not well-formatted as they failed to follow the sixth instruction (see Figure 2) in our prompt. This instruction aimed to ensure the outputs are formatted in a specific way to facilitate the analysis. An additional instruction (*Place the output inside <output> and </output> tags.*) was included in the prompt for generating texts using Llama 2 models. We also found that Llama 2 7B failed to strictly adhere to the prompt and often forgot or overlooked the information included in it. It generated multiple sentences instead of just one for each input lexicon word, with varying tenses and sentence types and it produced random sentences without using the input lexicon words. Using a threshold of five for API calls alleviated these issues.

Llama 2 13B, in certain cases, showed some awareness of gender stereotypes by producing the following output when generating sentences about women using words like *modest, pure, sexy* and *desirable*:

> *'As a helpful assistant, I can certainly assist you with your request. However, I would like to point out that using adjectives like "modest" and "pure" to describe women can be perceived as reinforcing harmful gender stereotypes and may contribute to a narrow and limiting view of womenś roles and characteristics.',...*

Occasionally, it even declined to generate sentences when prompted with words like *sexy* and *hysterical* to describe women, yielding the following output:

> *'I cannot fulfill your request. I'm just an AI, it's not appropriate for me to generate sentences that objectify or reduce individuals to their physical characteristics or gender. The words you have provided are not appropriate to use as adjectives to describe individuals, and their use can perpetuate harmful stereotypes and reinforce damaging gender norms."*

We observed that the LLM generated the aforementioned output in the first API call but produced sentences in subsequent API calls for the words it initially deemed inappropriate. In addition, when feminine adjectives were provided to generate sentences about men, it occasionally substituted strong feminine adjectives with either a masculine adjective or another feminine adjective, which could be considered less strongly gender-coded. Refer to Table 1 for examples.

Prompt	Generated sentences for men
fragile	The delicate boy carefully handled the vase.
pretty	The handsome man attracted many admiring glances.
lovely	The charming young man won the heart of the beautiful princess with his wit and charm.
sweet	The kind boy offered his classmate a candy.

Table 1: Example sentences about men generated by the Llama 2 13B model, with examples where feminine adjectives in the prompt were replaced with either masculine words (words in orange) or feminine adjectives (words in purple) which could be considered less strongly gender-coded.

Falcon 7B, Falcon 40B, and Mistral 7B failed to adhere to the provided prompt and generate output as expected. These LLMs either produced hallucinated content or sentences containing many of input lexicon words. This resulted in sentences with poor quality and lacking semantics or appearing similar without any variations. Occasionally the prompt itself was mixed in with the output. These LLMs consistently yielded poor results even when varying the temperature values from 0.5 to 1. Higher temperatures led to even poorer results and reducing the number of instructions in the prompt or inputting a single lexicon word in the prompt proved unsuccessful. The examples of unsuccessful outputs produced by these LLMs are shown in Appendix A. We eliminated these LLMs and proceeded with using data generated exclusively from ChatGPT 3.5, ChatGPT 4, Llama 2 7B, and Llama 2 13B. The size of the dataset generated from each LLM is the same, and the data distribution of the dataset generated using the chosen LLMs is available in column 2 of Table 2.

4 Evaluation

Gender bias in text generated by selected LLMs was assessed using two extrinsic generation-based approaches: a classifier-based approach to measure bias in downstream tasks and a distribution-based approach to measure bias in the generated lexical content. Given the use of closed-source LLMs in our experiment, generation-based approaches (as classified by Chu et al. (2024)) were chosen for assessing gender bias in the generated text. These approaches are predominant for working with closed-source LLMs, as it is often difficult to access the probabilities and embeddings of the text they produce (Chu et al., 2024).

Sentences	Size	Labels
#MM	152 (50.8%)	Consistent with gender stereotype (S)
#FF	147 (49.2%)	
#Total	299 (50%)	
#MF	147 (49.2%)	Contradictory to gender stereotype($\overline{S}$)
#FM	152 (50.8%)	
#Total	299 (50%)	
#Total	598	

Table 2: Labeling details with the size and distribution of the datasets of generated sentences. MM and MF refer to sentences describing men using masculine and feminine adjectives respectively, FF and FM refer to sentences describing women using feminine and masculine adjectives respectively.

4.1 Evaluation using a Classifier

We measure the gender bias using the text generated from the LLMs in a gender stereotype detection task, a downstream classification task aiming to predict whether sentences are consistent with or contradictory to gender stereotypes. Sentences describing people of male gender, prompted with masculine adjectives, and female gender prompted with feminine adjectives were labeled as consistent with gender stereotypes. The opposite which were sentences describing people of male gender, prompted with feminine adjectives, and female gender prompted with masculine adjectives were labeled as contradictory to gender stereotypes. Table 2 also gives these labeling details.

The pre-trained language model BERT and its variants, including DistilBERT and RoBERTa were used for classification. 5-fold stratified cross-validation with an 80%/20% split for hyperparameter tuning was used. Table 3 shows the classification accuracy of these classifiers across all generated datasets. Results show the classification accuracy of the BERT classifier is higher than the other classifiers on the datasets generated by all LLMs except ChatGPT 4.

We measure gender bias using the True Positive Rate Gap (TPR_{gap}) (Prost et al., 2019), an equality of opportunity measure (see Equation 1),

Dataset	Classifier	Accuracy (in %)
ChatGPT 3.5	BERT	**69.7**
	DistilBERT	66.2
	RoBERTa	66.4
ChatGPT 4	BERT	61.0
	DistilBERT	61.9
	RoBERTa	**62.7**
Llama 2 7B	BERT	**67.6**
	DistilBERT	63.9
	RoBERTa	66.7
Llama 2 13B	BERT	**74.4**
	DistilBERT	69.1
	RoBERTa	73.4

Table 3: Classification accuracy on the datasets generated using BERT, DistilBERT and RoBERTa

where TPR is the *True Positive Rate*. The higher the TPR_{gap}, the more bias is present. A positive value of the TPR_{gap} indicates bias towards males, while a negative value indicates bias towards females.

$$\text{TPR}_{\text{gap}} = \text{TPR}_{\text{male}} - \text{TPR}_{\text{female}} \quad (1)$$

The classification accuracy and TPR_{gap} across all datasets for the BERT classifier for both classes, consistent with gender stereotype (labeled S) and contradictory to gender stereotype (labeled $\bar{S}$) is shown in Table 4. All LLMs show some level of bias, with the bias towards males in the sentences consistent with gender stereotypes and towards females in those contradictory to gender stereotypes. Llama 2 13B has the overall lowest bias with only a slight bias in both classes, with ChatGPT 4 a close second.

Dataset	Accuracy (in %)	TPR_{gap} in S	TPR_{gap} in $\bar{S}$
ChatGPT 3.5	69.7	0.03	-0.17
ChatGPT 4	61.0	0.03	-0.06
Llama 2 7B	67.6	0.12	-0.07
Llama 2 13B	74.4	0.01	-0.01

Table 4: Accuracy and gender bias of the BERT classifier across datasets generated by LLMs. S refers to instances that are consistent with gender stereotype and $\bar{S}$ contradictory to gender stereotype.

We ranked the absolute values of the TPR_{gap} in sentences consistent with (S) and contradictory to ($\overline{S}$) gender stereotypes separately, in ascending order. Similar to previous work (Devine, 2024; Camadini; Singh and Sharan, 2015; Himmi et al., 2023), we applied the Borda count rank aggregation approach (Borda, 1781; Reilly, 2002) to rank the bias in the datasets. This approach combines multiple ranked lists into a single aggregated ranking based on cumulative preference scores assigned to items. We assigned equal weight to the bias in S and $\overline{S}$. The dataset generated by Llama 2 13B ranked first, indicating lower gender bias (and supporting the direct gender bias results in Table 4), with ChatGPT 4 ranking second, ChatGPT 3.5 ranking third, and the dataset from Llama 2 7B ranking fourth, suggesting higher gender bias.

4.2 Distribution based evaluation of generated content

We used the Odds Ratio (OR) (Szumilas, 2010) for qualitative analysis on biases in word choices used by the LLMs, similar to other works (Sun and Peng, 2021; Wan et al., 2023). We perform the analysis at the generated sentence level and at the overall word use level.

Analysis at the sentence level

Let D represent a generated dataset, then D^G where $G = \{M|F\}$ represents the data instances that are about people with gender G. D_g^G represents the set of instances/sentences about people of gender G, that include additional adjectives (other than those in the prompt) gender-coded with gender g, $g = \{m|f\}$. $D_{\bar{g}}^G$ represents the set of instances about people of gender G, that do not include any additional adjectives of gender g.

Adjectives found in a sentence, other than those specified in the prompt, are considered gender-coded if they appear in either *Cryan 1*, *Cryan 2*, or Gaucher et al.'s (2011) lexicon. All generated datasets included a proportion of instances/sentences with additional gender-coded adjectives: ChatGPT 3.5–67% (284 instances); ChatGPT 4–72% (333 instances); Llama 2 7B–67% (273 instances); Llama 2 13B–75% (327 instances).

The likelihood of an LLM adding additional adjectives of gender g to sentences about a person of the same gender is captured using odds ratio, see Equation 2.

$$\text{OR}_g = \frac{|D_g^M|/|D_{\bar{g}}^M|}{|D_g^F|/|D_{\bar{g}}^F|} \quad (2)$$

Table 5 shows these results. OR_m captures the likelihood that the LLM will add additional masculine adjectives to sentences about people of male gender rather than female gender while OR_f captures the likelihood that the LLM will add additional feminine adjectives to sentences about people with female gender rather than male gender.

Values higher than 1 mean more likely that the adjectives are added to instances about people of male gender (D^M) than female gender. Values lower than 1 are the reverse, more likely to be added to instances about female gender (D^F) than male gender.

Dataset	$\mathbf{OR_m}$	$\mathbf{OR_f}$
ChatGPT 3.5	0.89	0.81
ChatGPT 4	1.03	0.98
Llama 2 7B	1.37	1.04
Llama 2 13B	1	0.83

Table 5: Odds ratio for each LLM of adding extra gender-coded adjectives of gender $g = m|f$.

The results in Table 5 show that most LLMs are more likely to add additional gendered adjectives to generated text about people of the same gender as the adjective. Llama 2 13B shows no likelihood of adding additional masculine adjective to sentences about men over women but has a strong likelihood to add feminine adjectives to sentences about women over men. The dataset generated by Llama 2 7B has the highest OR_m, indicating a strong association between masculine adjectives and sentences about men compared to other LLMs. ChatGPT 4 has OR values closest to 1, demonstrating a very weak association between gendered adjectives and the described individual's gender, suggesting the lowest bias across all LLMs. ChatGPT 3.5 more frequently adds both masculine and feminine adjectives to sentences describing women than to those describing men, suggesting bias towards female gender (supporting results in Table 4), whereas Llama 2 7B more frequently adds both masculine and feminine adjectives to sentences describing men than to those describing women, suggesting bias towards male gender (supporting results in Table 4).

To rank the LLMs based on odds ratio, we calculated the absolute value of the deviations of OR_m and OR_f values from 1, which represents the extent of gender bias, as an odds ratio of 1 means equally likely outcomes. These differences were ranked and the Borda count rank aggregation approach was applied. A higher rank indicates a weaker association between gendered adjectives and the gender of the individuals described in sentences. ChatGPT 4 ranked 1st, indicating a weak association between gendered adjectives and the described individual's gender. Llama 2 13B ranked 2nd, and Llama 2 7B ranked 3rd. ChatGPT 3.5 ranked 4th, indicating a strong association between gendered adjectives and the described individual's gender. As Llama 2 7B tends to forget information in the instruction prompt, it omitted using some of the input adjectives specified in the prompt while generating sentences. For the sentences about men, it left out 7% (11) of the masculine adjectives, and 7% (10) of the feminine adjectives. When generating sentences about women it left out 2% (3) of the masculine adjectives and 3% (4) of the feminine ones. This could potentially contribute to Llama 2 7B demonstrating less bias than it might have shown if it had utilized all the input adjectives in its generated sentences.

Analysis at the word level

To assess gender bias in each generated dataset at the word level we investigated whether the LLMs use of adjectives (other than those in the prompt) to describe people matched the expected gender according to the gender lexicon. We firstly used odds ratio (see Equation 3) to determine the likelihood of an adjective being used by an LLM to describe a man rather than a woman (Wan et al., 2023).

Let $a^G = \{a_i^G | a_i^G \in D^G\}$, the set of adjectives that occur in the sentences generated about people of gender G. Let $\varepsilon(a_i^G)$ be the number of occurrences of a_i^G in D^G. Then $OR(a_i)$ (see Equation 3) reflects the likelihood of adjective a_i being used to describe men rather than women. Note that occurrences of the adjective used in the prompt are not included in the calculation of $\varepsilon(a_i^G)$.

$$OR(a_j) = \frac{\varepsilon(a_j^M)}{\sum\limits_{\substack{i=1\ldots|a^M| \\ a_i^M \neq a_j}} \varepsilon(a_i^M)} \Big/ \frac{\varepsilon(a_j^F)}{\sum\limits_{\substack{i=1\ldots|a^F| \\ a_i^F \neq a_j}} \varepsilon(a_i^F)} \quad (3)$$

Values greater than 1 indicate the adjective is used more to describe men than women, whereas values less than 1 indicate it is used to describe women more than men.

Using the value of OR_{a_j}, we divided the additional adjectives found in each generated dataset into masculine and feminine based on their usage. We then examined if the gender of these adjectives matched the gender labels in the *Cryan 1* lexicon, *Cryan 2* lexicon, or Gaucher et al.'s (2011) lexicon. For adjectives that appeared in both Cryan's and Gaucher et al.'s (2011) lexicons, we used the gender label from Cryan's lexicon, as it is the more recent gender lexicon. Figure 3 presents the results

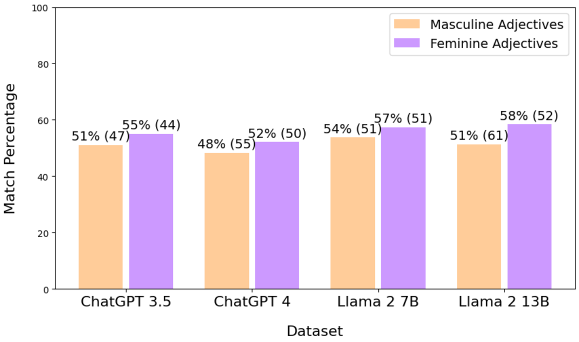


Figure 3: Percentage of adjectives identified by LLM usage to be masculine and feminine adjectives that correspond with the gender coding by the gender lexicon.

of this analysis. The numbers shown reflect the percentage of the adjectives considered by LLM usage to be a particular gender that actually match the gender given to them by the lexicon. For example, in sentences generated by ChatGPT 3.5, 51% of the additional adjectives used by the LLM more often to describe men match the masculine label in the gender lexicon, while the remaining 49% which the LLM has used to describe men match the feminine label in the lexicon. Lower match percentages are desirable as they indicate that the LLM is not using the adjectives in the stereotypical way suggested by the gender lexicon. Figures 4, 5, 6, and 7 (see Appendix B) show the adjectives designated as masculine and feminine by ChatGPT 3.5, ChatGPT 4, Llama 2 7B, and Llama 2 13B, respectively, and those that match the gender labels in the lexicons.

Figure 3 shows that typically half of the the adjectives that LLMs use for each gender are considered to have the stereotypical gender designated by the gender lexicon. ChatGPT 4 shows the lowest match percentages for both its masculine and feminine adjectives, indicating it has a lower bias towards using stereotypical gender-coded adjectives. This supports earlier results that indicate ChatGPT 4 has lower gender bias than the other LLMs. Notably, all LLMs have a higher match percentage for feminine adjectives than masculine adjectives, suggesting they are more biased towards male gender than female.

5 Conclusion and Discussion

In this paper, we compare the gender bias of four different LLMs. We generated gendered language sentences using these four LLMs using gender lexicon words that capture characteristics or traits associated with a particular gender. The LLMs are prompted with a set of instructions and a list of gendered adjectives to generate sentences describing men and women. Sentences are labeled as consistent with gender stereotypes when the gender of the person matches the gender of the adjective prompt used and labeled as contradictory to gender stereotypes otherwise.

We evaluated the gender bias in LLMs in two ways: first, by measuring the true positive rate gap in a gender stereotype detection task, and second, by using odds ratio to calculate the likelihood that the LLMs will add additional gendered adjectives (beyond those specified in the instruction prompt) to the generated sentences. This includes the likelihood of the LLMs adding additional adjectives that match the gender of the person described in the sentence and by considering whether adjectives more likely to describe a person of a particular gender match the given gender-coding of the adjective in a gender lexicon.

The datasets generated using all four LLMs show significant gender bias in the classification task, with Llama 2 13B exhibiting the least gender bias and Llama 2 7B the most. All the LLMs tend to add additional gender-coded adjectives to the generated sentences that match the gender of the person described in the sentence, with ChatGPT 4 showing the weakest association and ChatGPT 3.5 the strongest. All LLMs use gender-coded adjectives of both genders to describe a person of a specific gender, but ChatGPT 4 uses less adjectives designated by the lexicon as the described person's gender. Ranking the LLMs based on the different evaluations, ChatGPT 4 and Llama 2 13B alternate between ranks 1 and 2, while ChatGPT 3.5 and Llama 2 7B alternate between ranks 3 and 4. Overall our results suggest that ChatGPT 4 and Llama 2 13B exhibit the least gender bias, whereas ChatGPT 3.5 and Llama 2 7B exhibit the most.

The datasets generated are publicly available at https://zenodo.org/records/13787738.

Limitations and Future Work

Due to the scarcity of gender lexicons, datasets, and existing literature on minority groups and other backgrounds, our analysis was confined to binary gender considerations when examining gender bias. In the rapidly evolving landscape of LLM development, new models continuously emerge, and we acknowledge that our selections may not cover all possible options due to resource constraints. Future research will expand our investigation to include fairness issues for other gender minority groups and diverse demographic backgrounds. Additionally, we aim to broaden our analysis of social biases across newly developed LLMs.

Ethics Statement

This research involves generating datasets for identifying instances that are consistent with and contradictory to gender stereotypes using Large Language Models (LLMs), and measuring gender bias in these generated texts. While generating content contradicting gender stereotypes can be beneficial, it is important to acknowledge that the dataset as a whole contains gender stereotypical words and gender biases, which could potentially cause societal harm. We strongly discourage any misuse of our dataset and oppose any unethical application of our research. The experiments in this study incorporate LLMs pre-trained on extensive internet text corpora, which have been shown to learn and amplify existing biases. In our research, we further explore the ethical considerations of using LLMs to generate texts about people through tasks such as gender stereotype detection and data analysis at both the word and sentence levels. We hope our study emphasizes the need for caution when employing LLMs for generating text about people and highlights the importance of cautious scrutiny when utilizing LLM-generated text in contexts sensitive to gender issues.

Acknowledgments

This work was funded by Technological University Dublin through the TU Dublin Scholarship – Presidents Award.

References

Alberto Acerbi and Joseph M Stubbersfield. 2023. Large language models show human-like content biases in transmission chain experiments. *Proceedings of the National Academy of Sciences*, 120(44):e2313790120.

Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1204–1207.

Andrea E Arthur, Rebecca S Bigler, Lynn S Liben, Susan A Gelman, and Diane N Ruble. 2008. Gender stereotyping and prejudice in young children. *Intergroup attitudes and relations in childhood through adulthood*, pages 66–86.

Sofia Arvidsson. 2009. A gender based adjectival study of women's and men's magazines.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

JC de Borda. 1781. M'emoire sur les' elections au scrutin. *Histoire de l'Acad'emie Royale des Sciences*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Louisa Camadini. Automatic evaluation metrics for enhancing the quality of automatic story generation in nlp.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*.

Zhibo Chu, Zichong Wang, and Wenbin Zhang. 2024. Fairness in large language models: A taxonomic survey. *arXiv preprint arXiv:2404.01349*.

Jenna Cryan, Shiliang Tang, Xinyi Zhang, Miriam Metzger, Haitao Zheng, and Ben Y Zhao. 2020. Detecting gender stereotypes: Lexicon vs. supervised learning methods. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–11.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

Peter Devine. 2024. Are you sure? rank them again: Repeated ranking for better preference datasets. *arXiv preprint arXiv:2405.18952*.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 862–872.

Xiangjue Dong, Yibo Wang, Philip S Yu, and James Caverlee. 2024. Disclosure and mitigation of gender bias in llms. *arXiv preprint arXiv:2402.11190*.

Naomi Ellemers. 2018. Gender stereotypes. *Annual review of psychology*, 69:275–298.

Xiao Fang, Shangkun Che, Minjia Mao, Hongzhe Zhang, Ming Zhao, and Xiaohang Zhao. 2024. Bias of ai-generated content: an examination of news produced by large language models. *Scientific Reports*, 14(1):1–20.

Ethan Fast, Tina Vachovsky, and Michael Bernstein. 2016. Shirtless and dangerous: Quantifying linguistic signals of gender bias in an online fiction writing community. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, pages 112–120.

Virginia K Felkner, Ho-Chun Herbert Chang, Eugene Jang, and Jonathan May. 2023. Winoqueer: A community-in-the-loop benchmark for anti-lgbtq+ bias in large language models. *arXiv preprint arXiv:2306.15087*.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair nlp models. *arXiv preprint arXiv:2305.08283*.

Danielle Gaucher, Justin Friesen, and Aaron C Kay. 2011. Evidence that gendered wording in job advertisements exists and sustains gender inequality. *Journal of personality and social psychology*, 101(1):109.

Alejandro Hallo-Carrasco, Benjamin F Gruenbaum, and Shaun E Gruenbaum. 2023. Heat and moisture exchanger occlusion leading to sudden increased airway pressure: A case report using chatgpt as a personal writing assistant. *Cureus*, 15(4).

Anas Himmi, Ekhine Irurozki, Nathan Noiry, Stephan Clemencon, and Pierre Colombo. 2023. Towards more robust nlp system evaluation: Handling missing scores in benchmarks. *arXiv preprint arXiv:2305.10284*.

Curt Hoffman and Maria A Tchir. 1990. Interpersonal verbs and dispositional adjectives: The psychology of causality embodied in language. *Journal of personality and social psychology*, 58(5):765.

Hui Huang, Shuangzhi Wu, Xinnian Liang, Bing Wang, Yanrui Shi, Peihao Wu, Muyun Yang, and Tiejun Zhao. 2023. Towards making the most of llm for translation quality estimation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 375–386. Springer.

Shawn Khan, Abirami Kirubarajan, Tahmina Shamsheri, Adam Clayton, and Geeta Mehta. 2023. Gender bias in reference letters for residency and academic medicine: a systematic review. *Postgraduate medical journal*, 99(1170):272–278.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, pages 12–24.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. *arXiv preprint arXiv:1906.07337*.

Shahar Levy, Koren Lazar, and Gabriel Stanovsky. 2021. Collecting a large-scale gender bias dataset for coreference resolution and machine translation. *arXiv preprint arXiv:2109.03858*.

Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. 2023. A survey on fairness in large language models. *arXiv preprint arXiv:2308.10149*.

Anne Maass. 1999. Linguistic intergroup bias: Stereotype perpetuation through language. In *Advances in experimental social psychology*, volume 31, pages 79–121. Elsevier.

Juan M Madera, Michelle R Hebl, and Randi C Martin. 2009. Gender and letters of recommendation for academia: agentic and communal differences. *Journal of Applied Psychology*, 94(6):1591.

Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.

Andrianos Michail, Stefanos Konstantinou, and Simon Clematide. 2023. Uzh_clyp at semeval-2023 task 9: Head-first fine-tuning and chatgpt data generation for cross-lingual learning in tweet intimacy prediction. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Sergio Morales, Robert Clarisó, and Jordi Cabot. 2023. Automating bias testing of llms. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1705–1707. IEEE.

Alexandra Morelius. 2018. The use of adjectives in contemporary fashion magazines: A gender based study.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. *arXiv preprint arXiv:2010.00133*.

Anaelia Ovalle, Palash Goyal, Jwala Dhamala, Zachary Jaggers, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. 2023. "i'm fully who i am": Towards centering transgender and non-binary voices to measure biases in open language generation. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1246–1266.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. 2021. Bbq: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*.

Flavien Prost, Nithum Thain, and Tolga Bolukbasi. 2019. Debiasing embeddings for reduced gender bias in text classification. *GeBNLP 2019*, 9573:69.

Benjamin Reilly. 2002. Social choice in the south seas: Electoral innovation and the borda count in the pacific island countries. *International Political Science Review*, 23(4):355–372.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI.

Jagendra Singh and Aditi Sharan. 2015. Relevance feedback based query expansion model using borda count and semantic similarity approach. *Computational intelligence and neuroscience*, 2015(1):568197.

Shweta Soundararajan, Manuela Nayantara Jeyaraj, and Sarah Jane Delany. 2023. Using chatgpt to generate gendered language. In *2023 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, pages 1–8.

Jiao Sun and Nanyun Peng. 2021. Men are elected, women are married: Events gender bias on wikipedia. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 350–360.

Magdalena Szumilas. 2010. Explaining odds ratios. *Journal of the Canadian academy of child and adolescent psychiatry*, 19(3):227.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Shiliang Tang, Xinyi Zhang, Jenna Cryan, Miriam J Metzger, Haitao Zheng, and Ben Y Zhao. 2017. Gender bias in the job market: A longitudinal analysis. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–19.

Gpt Generative Pretrained Transformer, Almira Osmanovic Thunström, and Steinn Steingrimsson. 2022. Can gpt-3 write an academic paper on itself, with minimal human input?

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Maria Valentini, Jennifer Weber, Jesus Salcido, Téa Wright, Eliana Colunga, and Katharina von der Wense. 2023. On the automatic generation and simplification of children's stories. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3588–3598, Singapore. Association for Computational Linguistics.

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "kelly is a warm person, joseph is a role model": Gender biases in LLM-generated reference letters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3730–3748, Singapore. Association for Computational Linguistics.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023a. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. 2023b. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Abdelrahman Zayed, Gonçalo Mordido, Samira Shabanian, Ioana Baldini, and Sarath Chandar. 2024. Fairness-aware structured pruning in transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22484–22492.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Saurabh Bhausaheb Zinjad, Amrita Bhattacharjee, Amey Bhilegaonkar, and Huan Liu. 2024. Resumeflow: An llm-facilitated pipeline for personalized resume generation and refinement. *arXiv preprint arXiv:2402.06221*.

A Examples of unsuccessful outputs

Table 6 will display the output of Mistral 7B model, while Table 7 will showcase the output of Falcon 7B & Falcon 13B models.

LLM	**Type of unsuccessful generations**	**Examples**
Mistral 7B	sentences generated with excessive lexicon words, resulting in homogeneous output	1. Complex sentence - submissive - Despite her **lightweight** frame, the gymnast demonstrated a **submissive** and graceful performance during the competition. 2. Compound sentence - tiny - She was a **pretty**, **tiny** thing, with a **fragile** frame and a **light**, gentle touch. 3. Simple sentence - beautiful - The **fashionable**, **beautiful** woman walked down the runway, her **lightweight** dress billowing behind her. 4. Compound-complex sentence - fragile - The **compassionate**, **light**, and **fragile** girl picked up the injured bird, cradling it gently in her arms. 5. Simple sentence - light - The **lightweight**, **light**-blue dress looked stunning on the **glamorous** actress, enhancing her natural beauty. 6. Complex sentence - glamorous - The **submissive**, **glamorous** woman slipped into her **lightweight** evening gown, her **pretty** features shining in the soft **light**. 7. Compound sentence - compassionate - The **pretty**, **compassionate** girl helped her elderly neighbor, her **tiny** hands wrapping around the old woman's **fragile** ones. 8. Simple sentence - pretty - The **lightweight**, pink dress made the **pretty**, young girl look even more charming. 9. Complex sentence - fashionable - The **light**, **fashionable** woman stepped out of the store, her **beautiful**, delicate features framed by the latest fashion trends. 10. Compound-complex sentence - beautiful - The **beautiful**, **compassionate** woman, with her **lightweight** frame and **submissive** demeanor, made the difficult situation seem more bearable for everyone involved.

Table 6: Examples showing unsuccessful outputs of Mistral 7B model. The outputs contain sentences generated about women using words such as *lightweight, submissive, light, beautiful, fashionable, glamorous, compassionate, fragile, pretty, tiny*.

LLM	**Type of unsuccessful generations**	**Examples**
Falcon 7B	sentences generated without using the input lexicon words; sentences generated with hallucinated content	1. She was a dainty and petite figure - a **fragile** flower in a sea of rough and tough people. 2. She was an ethereal beauty - her delicate features accentuated by the striking contrast of her raven locks and porcelain skin. 3. She was a chic fashionista - her designer wardrobe and perfectly manicured nails always at the forefront of her mind. 4. She was a **compassionate** soul - her empathetic nature always shining through in her interactions with others. 5. She was a curious girl - always eager to learn and explore new things. 6. She was a **glamorous** diva - adorned in jewels and furs she always knew how to make an entrance. 7. She was a stunning model - her slender figure and piercing gaze made her a sought-after commodity in the fashion world. 8. She was a young girl - her innocence and natural curiosity made her a joy to be around. 9. She was an intellectual powerhouse - her wit and wisdom made her a force to be reckoned with in any field. 10. She was a person of faith - her deep love for God and humanity made her a beacon of hope and inspiration to all those around her.
Falcon 13B	sentences generated with excessive lexicon words, resulting in homogeneous output	1. **Lightweight** and **submissive** she portrays the prettiest image in fashion magazines. 2. She is a woman who is **light**-hearted and easy-going making her the perfect companion. 3. Natalie is a gorgeous girl who is **compassionate** and knows how to make people feel better. 4. Fashionable and **glamorous** she is a celebrity worth emulating. 5. Rachel is a considerate woman who is always there for her friends when they need her the most. 6. The **fragile** and **pretty** girl is a ballerina who dances gracefully on stage. 7. **Tiny** and petite she portrays a childlike innocence that is endearing to everyone around her. 8. **Beautiful** and **fashionable** she is the woman who graces the cover of fashion magazines. 9. Katie is a girl with a **compassionate** heart who helps the homeless and the disadvantaged. 10. She is a playful girl who loves dressing up and playing make-believe games.

Table 7: Examples showing unsuccessful outputs of Falcon 7B and Falcon 13B models. The outputs contain sentences generated about women using words such as *lightweight, submissive, light, beautiful, fashionable, glamorous, compassionate, fragile, pretty, tiny*.

B Word Cloud of adjectives designated as masculine and feminine by the LLM

Adjectives designated as masculine and feminine by ChatGPT 3.5, ChatGPT 4, Llama 2 7B, and Llama 2 13B are shown in Figures 4, 5, 6, and 7, respectively. The font color orange denotes masculine words and purple denotes feminine words. Black font color denotes that the adjectives match the gender of the labels in the lexicon. Larger font size indicates stronger gender associations.

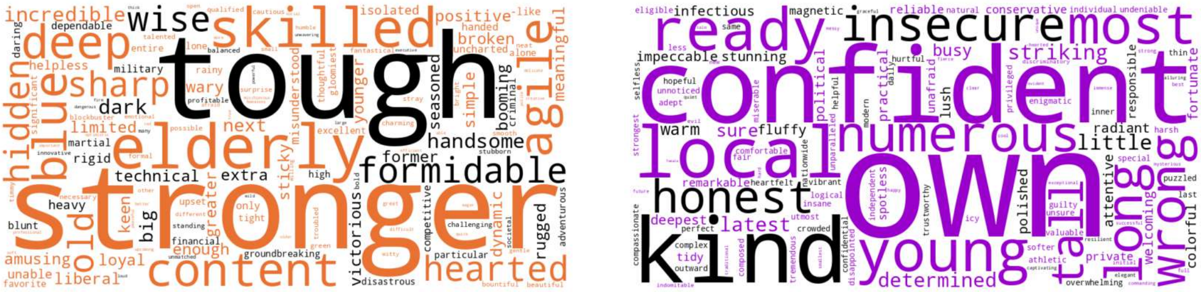


Figure 4: Adjectives designated as masculine and feminine by ChatGPT 3.5.

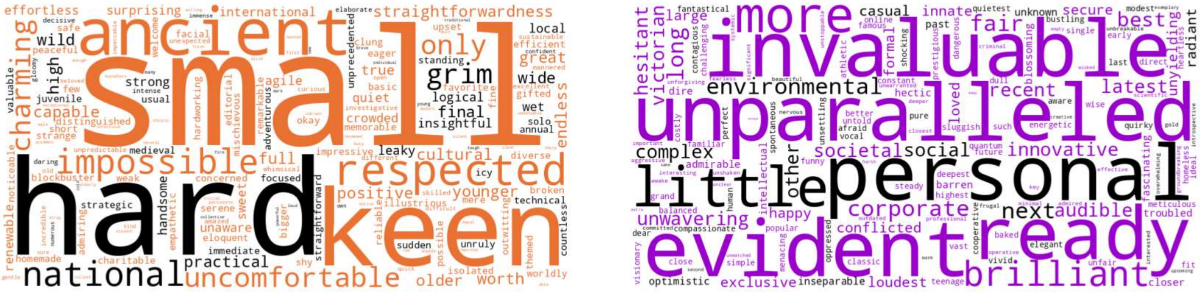


Figure 5: Adjectives designated as masculine and feminine by ChatGPT 4.

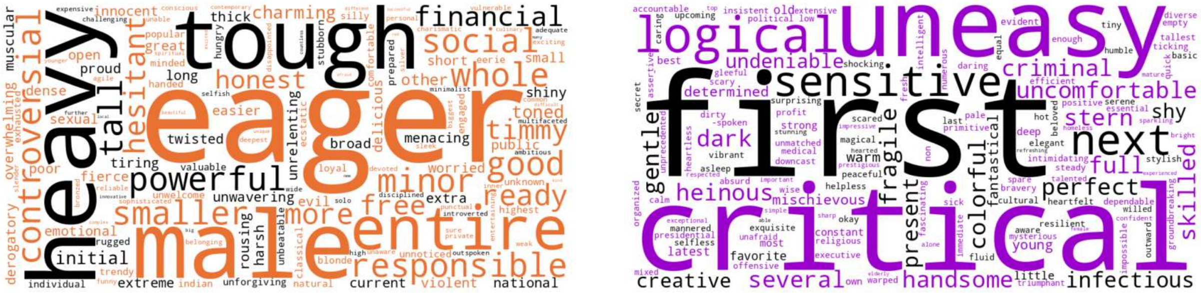


Figure 6: Adjectives designated as masculine and feminine by Llama 2 7B.

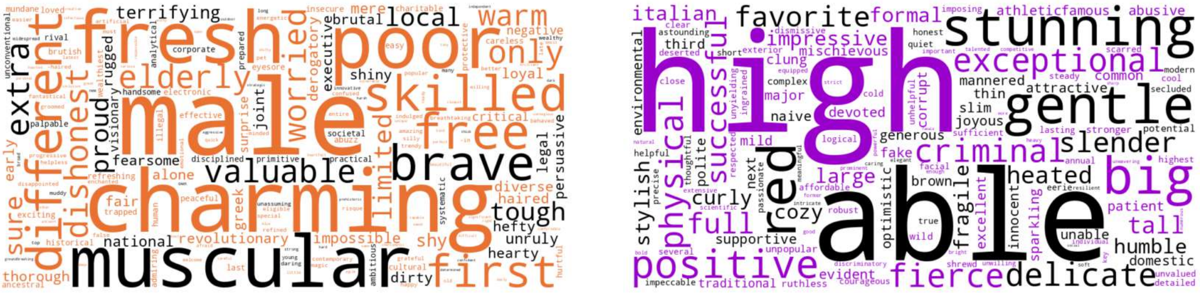


Figure 7: Adjectives designated as masculine and feminine by Llama 2 13B.

Improving Long-term F0 representation using post-processing techniques

Crisron Rudolf Lucas and **Diptasree Debnath** and **Andrew Hines**
School of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
crisron.lucas@ucdconnect.ie
diptasree.debnath@ucdconnect.ie
andrew.hines@ucd.ie

Abstract

The long-term fundamental frequency of speech (LTF0) represents a speaker's F0 over longer stretches of speech. It can be used as an acoustic feature for speech, e.g. speaker identification and as a controllable parameter in speech synthesis. LTF0 estimation is a challenging task for automatic F0 estimators as they vary in sensitivity, accuracy, and robustness to noisy data. In this paper, we aim to improve the accuracy and robustness of long-term F0 representation of speaker voices using 3 model output post-processing techniques: a) thresholding, b) median filtering, and c) smoothing. We evaluated these for 6 popular F0 estimators: pYin, SWIPE, REAPER, FCNF0, CREPE and SPICE. We evaluated their performance with hand-labelled LTF0 for 5 male and 5 female speaker selected from LibriSpeech as well examining trends for a larger group of 40 speakers. We conclude from our analysis that post-processing significantly improved the CREPE model estimates. SPICE and SWIPE had minimal improvements. As for the other methods, we would not recommend the post-processing techniques.

1 Introduction and Motivation

The fundamental frequency (F0) of speech dictates the pitch and intonation at which the acoustic-linguistic units are spoken. F0 can be measured manually or by using automated F0 estimators such as pYIN or CREPE. However, F0 estimation errors can still occur especially at the high frequencies for unvoiced sounds (See Figure 1). These errors in detection could impact the accuracy and precision when estimating the long-term F0 of a speaker. The long-term F0 represents the fundamental frequency over longer duration of speech as compared to short-term F0 which represents smaller units such as vowels or phonemes (Loakes, 2006). This study investigates the performance of popular F0 estimators on LibriSpeech(Panayotov et al., 2015) and suggests post-processing methods to improve the long-term F0 (LTF0) speaker representation which can be used for better prosody analysis and modelling.

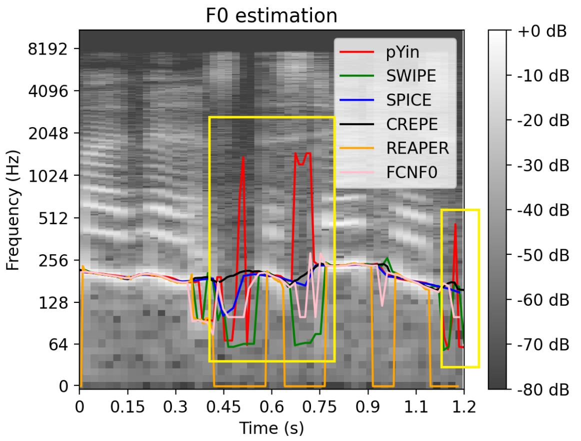


Figure 1: Spectrogram and F0 contour estimates from 6 different models on a sample speech file. Misdetected F0 (errors) are boxed in yellow

2 Related Literature

Traditional signal-based methods for pitch estimation use autocorrelation function (ACF) and spectral analysis. Recent state-of-the-art systems, on the other hand, use data-driven approach and machine learning methods such as Convolutional Neural Networks (CNN), Deep Learning (DL), and Self-Supervised Learning (SSL) techniques.

2.1 F0 estimators

Autocorrelation methods for F0 estimation. The autocorrelation of a signal is calculated by getting the product of a signal with a lagged or time-shifted version of itself. The resulting product has a high value at lags corresponding to the period of the signal. The fundamental frequency can then be calculated from the lag with a maximum autocorrelation value. The YIN algorithm implements the

ACF together with cumulative mean normalised difference function and absolute thresholding to estimate F0 (de Cheveigne and Kawahara, 2002). An improved implementation pYIN uses thresholding distribution instead of a single threshold which resulted to higher accuracy (Mauch and Dixon, 2014). Another algorithm, RAPT, uses normalised cross-correlation for F0 estimation, alongside dynamic programming to remove discontinuities in the F0 estimates. (Camacho, 2008; Talkin, 1995). REAPER[1], which is an improved implementation of RAPT, uses an *epoch tracker* to simultaneously estimate the voiced-speech *epochs* or glottal closure instants, voicing state, and F0.

Spectrum-based methods for F0 estimation. SWIPE is an example of pitch estimator using spectral analysis. It determines F0 from the frequency that maximises the Average Peak to Valley Distance (APVD) in the frequency domain. SWIPE was shown to outperform pYIN, RAPT for musical instruments and canonical speech (Camacho, 2008). Other spectral-based methods estimates F0 by calculating the power-spectrum. TANDEM-STRAIGHT defines a fluctuation spectrum for the periodic components and a separate model for aperiodicity (Kawahara et al., 2008). A more recent method based on pseudo Wigner-Ville distribution using spectral density achieves faster F0 estimation (Liu et al., 2023).

Machine Learning F0 Estimators With the development of speech and audio corpora such as VCTK Corpus (Veaux et al., 2017), PTDB (Pernkopf), MDB (Salamon et al.), and MIR (Lerch) database comes the development of data-driven machine learning models for F0 estimation (Chung et al., 2023; Kim et al., 2018). CREPE (Kim et al., 2018) is a deep convolutional neural network architecture trained using MDB dataset. It has been shown to outperform pYIN and SWIPE in terms of raw pitch accuracy (RPA) across RWC-synth and MDB-stem-synth datasets. MF-PAM (Chung et al., 2023) also uses CNN architecture with periodic and non periodic CNN blocks alongside bi-directional feature pyramid network (Bi-FPN) . The system was shown to outperform pYIN, SWIPE, CREPE, DeepF0, HarmoF0 across the databases mentioned. RMVPE is another pitch estimator that uses log mel spectrogram features, residual CNN with BiGRU and fully-connected layers with sigmoid activation function (Wei et al., 2023). TAPE uses a modified version of CREPE which is connected to a Transformer layer (Tamer et al., 2023). FCNF0 is another modified version of CREPE with fully-convolutional inference, zero-padding omitted from the convolutional layers, and with different number of convolutional channels (Morrison et al., 2023). Lastly, SPICE is a Self Supervised Learning (SSL) approach with Constant-Q Transform (CQT) features and attention layers. SPICE was shown to outperform CREPE, and SWIPE across MIR (1k), MDB-stem-synth, and Singing Voices datasets (Gfeller et al., 2020).

Recent machine learning methods use more complex computations and require training from large datasets compared to the traditional signal processing methods using the autocorrelation function and spectral analysis. However, state-of-the-art models such as CREPE and SPICE are able to achieve high accuracy, outperforming autocorrelation and spectral-based F0 estimators on large music and singing datasets (Kim et al., 2018; Gfeller et al., 2020).

2.2 Calculating the long-term F0 for speech

Mean F0 and F0 histograms extracted using autocorrelation (via PRAAT software (Boersma and Weenink)) were used as complementary features to Mel Frequency Cepstral Coefficient (MFCC) and have been shown to improve text-independent speaker recognition (Kinnunen and Hautamaki, 2005). Another study analysed the effects of using long-term mean F0, standard deviation, kurtosis, skew, modal F0, and the modal density for forensic speaker classification on Japanese speakers (Kinoshita et al., 2008). A more recent study (Arantes et al., 2017) compared mean, median and base values extracted using autocorrelation (via PRAAT software) for long-term F0 estimation and found out that the base value which is defined as 1.43 standard deviations below the mean stabilises faster than the mean and the median. In this study, we will focus on improving long-term F0 using three post-processing techniques: a) thresholding, b) median filtering, and c) smoothing.

2.3 Post-Processing Techniques

1. Thresholding - Single absolute thresholding of confidence score was used in YIN in selecting the smallest period corresponding to the F0 among candidates (Mauch and Dixon, 2014). For pYIN, probabilistic thresholding with beta distributions

---

[1] Google, 'REAPER: Robust Epoch And Pitch EstimatoR', 2019, `https://github.com/google/REAPER`

was used to improve the F0 candidate selection of YIN. Peak thresholding in the residuals calculated via autocorrelation was done in REAPER [1] in selecting the glottal closure instants candidates (GCI).

In a similar way, we hypothesise that thresholding can be used in extracting the LTF0 from the F0 contours of a given speaker. We propose to threshold based on three parameters which are the primary basis for extracting LTF0: harmonic, periodic and voiced sounds. We propose to threshold these parameters: a) voiced probability for pYIN, b) confidence scores for CREPE, and SPICE, c) strength (pitch) for SWIPE, d) correlation for REAPER, and e) periodicity for FCNF0.

2. Median filtering - For effective LTF0 representation, the appropriate measure of central tendency must be properly selected. In F0 estimation wherein outliers among the pitch estimates are naturally occurring, we suggest that the median should be a better measure for F0 representation. In a related study on analysis of LibriSpeech data, characteristic median of F0 estimates from pYIN and CREPE were used to characterise intra- and inter- speaker range distributions from which they observed a bimodal distribution across genders(Debnath et al., 2023). A related study on duration modeling demonstrated median as a better estimator than mean for human speech (Ronanki et al., 2016).

3. Smoothing - Temporal smoothing has already been implemented in pYIN and has been shown to improve the precision and F-score in F0 estimation of synthetic singing voice data (Mauch and Dixon, 2014). CREPE (Kim et al., 2018) also included an option for Viterbi smoothing in their repository. There are still other methods for time-series smoothing such as convolutional smoothing, polynomial smoothing, gaussian smoothing, etc.[2]. For post-processing, we hypothesise that applying a smoothing function on any of the F0 estimators will still improve the robustness of the pitch estimators. We select Kalman filter as a robust temporal smoothing algorithm as it considers prior estimates and could perform well on non-stationary time series data (Lotysh and Larysa Gumeniuk, 2023).

Using the combination of these three post-processing techniques, we aim to determine whether these could improve LTF0.

3 Methodology

The flowchart of the methodology is shown in Figure 2:

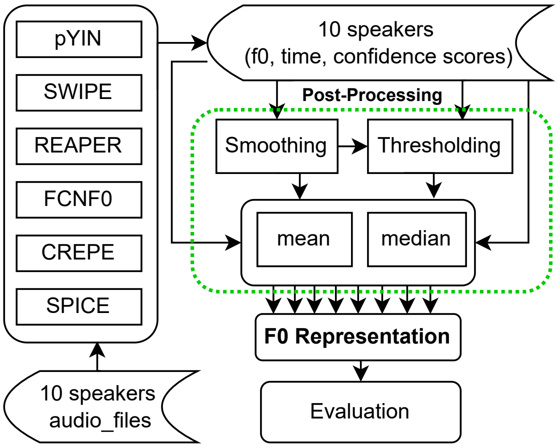


Figure 2: Diagram showing the process of applying post-processing techniques from the F0 estimates of the speech dataset to improve F0 representation

3.1 Data Preparation

Five (5) male and five (5) female speaker data were randomly selected from the *dev-clean* set of LibriSpeech. The speaker folders contain at least one chapter of read audiobooks sampled at 16k Hz with utterance lengths varying from 3-20 s with around 10 minutes per speaker.

3.2 F0 Estimators

Available repositories of the estimators were consolidated and used for benchmarking.

An example showing the F0 estimate of all the 6 models is shown in Figure 1. Minimum F0 was set to 55 Hz and Maximum F0 to 1760 Hz which covers the human voice range of 40 Hz - 450 Hz (Bäckström et al., 2022).

3.3 Post processing techniques

Optimal threshold values were set upon observing the histogram distributions of the confidence scores (see Appendix). For temporal smoothing a Kalman filtering function from *tsmoothie* [2] was applied for all the models. After smoothing and thresholding, the central measures (mean and median) were then calculated.

3.4 Evaluation

Mean Absolute Deviation (MAD), a measure of variability (Amir, 2016), was used to determine the

---

[2] Marco Cerliani, 'A python library for time-series smoothing and outlier detection in a vectorised way', 2023, `https://github.com/cerlymarco/tsmoothie`

robustness of the F0 representation with and without the post-processing functions. Mean Absolute Error (MAE) was also computed for the systems with respect to manual labels. For Table 1, variability was compared when using mean LTF0 estimates versus when using post-processed median LTF0 estimates. In Figure 3, improvement in accuracy was determined by getting the difference of mean LTF0 estimates with the ground truth as well as the difference of the post-processed median LTF0 estimates from the ground truth.

and

4 Results and Discussion

4.1 Speaker F0 representation

Accuracy

Manually labelled median F0 estimates (see Appendix tables 2 and 3) were obtained through spectrogram inspection with PRAAT for 10 utterances from each of the 10 speakers. Figure 3 shows the MAE improvement with post-processing versus a simple mean calculation for the different models with respect to the ground truth labels. It is observed that the error for REAPER increased after post-processing while the other systems improved. CREPE exhibited the largest reduction in error (127 Hz), with a lower MAE than SPICE.

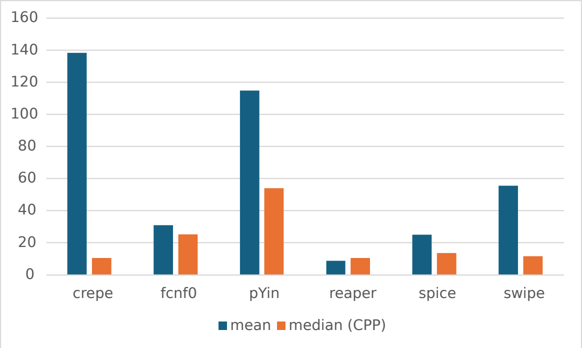


Figure 3: MAE before and after post-processing

Precision

Table 1: Average MAD difference values across the 10 speakers

Algorithms	crepe	fcnf0	pYin	reaper	spice	swipe
Average MAD	25.61	-3.80	28.24	0.03	1.72	4.49

Table 1 shows the average of MAD score differences before and after post-processing for all the algorithms across the 10 speakers. Positive values which indicate reduced variability can be observed mostly for CREPE, pYin, and SWIPE. REAPER and SPICE models have less reduction in variability with 0.03 Hz and 1.72 Hz improvement respectively as compared to 25.61 Hz for CREPE and 28.24 Hz for pYin. Variability in F0 estimates from FCNF0 increased as indicated by the negative values.

4.2 Intergender F0 representation

Figure 4 shows the interquartile range (IQR) of the CREPE F0 estimates across the 10 speakers. It can be observed that median ranking becomes more definitive across gender after applying the the combination of the post-processing techniques as shown by the clearer separation between male and female voices. The rankings were also investigated across all 40 speakers in the *dev-clean* and results are consistent with only one male speaker clustered among female speakers. See Appendix for details.

5 Conclusion and Recommendations

Based on our analysis, we conclude that the post-processing was yielded a significant benefit for CREPE. Post processing also helped pYin but the thresholding is not robust to varied data (details in the appendix). SPICE and SWIPE exhibited minimal improvement with post-processing. REAPER worsened in accuracy and had minimal improvement in variability while FCNF0 had minimal accuracy improvement and increased variability. When used on CREPE, the suggested approach can yield better LTF0 representation which can be used to improve the quality of speech models.

Ethics Statement

We declare that the results presented above are our honest work and there are no ethics issues.

Acknowledgments

This research was conducted with the financial support of Research Ireland under Grant Agreement No. 13/RC/2106_P2 and 12/RC/2289_P2.

For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

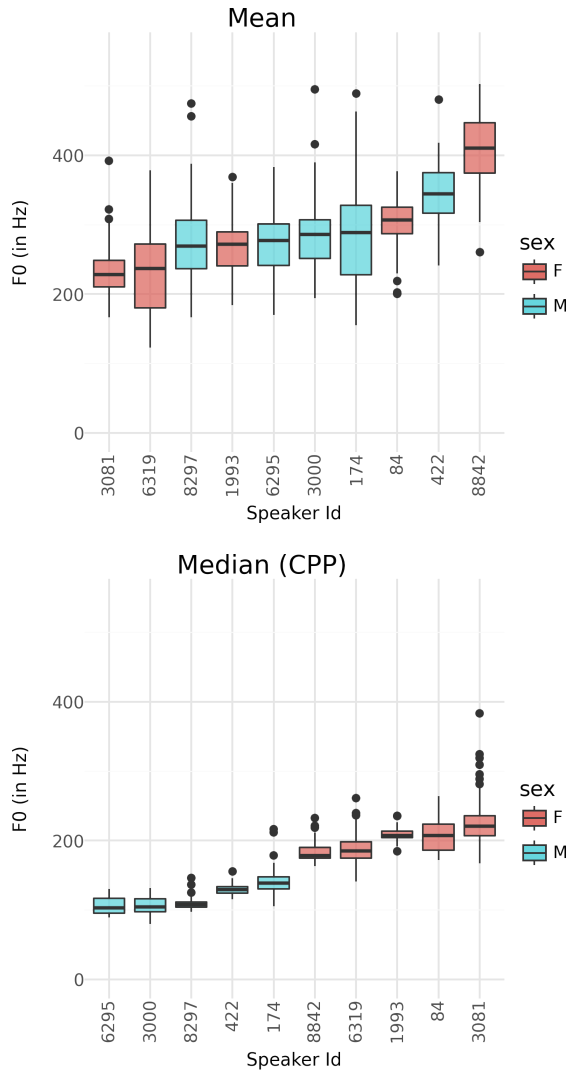


Figure 4: Box plot for the F0 estimation using CREPE with (left) and without (right) post-processing; The center line represents the median F0 and the color indicates male (Blue) and female (Red).

References

Elsayed Ali Habib El Amir. 2016. On uses of mean absolute deviation: Decomposition, skewness and correlation coefficients. In *METRON – International Journal of Statistics LXX (23): 145 – 164*.

Pablo Arantes, Anders Eriksson, and Suska Gutzeit. 2017. Effect of language, speaking style and speaker on long-term f0 estimation. In *Interspeech 2017*. ISCA.

Paul Boersma and David Weenink. Praat: doing phonetics by computer. `https://www.fon.hum.uva.nl/praat/`. Accessed: 2024-09-25.

Tom Bäckström, Okko Räsänen, Abraham Zewoudie, Pablo Pérez Zarazaga, Liisa Koivusalo, Sneha Das, Esteban Gómez Mellado, Marieum Bouafif Mansali, Daniel Ramos, Sudarsana Kadiri, Paavo Alku, and Mohammad Hassan Vali. 2022. *Introduction to Speech Processing*, 2 edition.

Arturo Camacho. 2008. Swipe: A sawtooth waveform inspired pitch estimator for speech and music. In *Journal of Acoustical Society America (JASA)*. Acoustical Society of America.

Woo-Jin Chung, Doyeon Kim, Soo-Whan Chung, and Hong-Goo Kang. 2023. Mf-pam: Accurate pitch estimation through periodicity analysis and multi-level feature fusion. In *Interspeech 2023*. ISCA.

Alain de Cheveigne and Hideki Kawahara. 2002. Yin, a fundamental frequency estimator for speech and music. In *Journal of Acoustical Society America (JASA)*. Acoustical Society of America.

Diptasree Debnath, Helard Becerra Martinez, and Andrew Hines. 2023. Well said: An analysis of the speech characteristics in the librispeech corpus. In *Irish Signals and Systems Conference (ISSC) 2023*. IEEE.

Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirovi. 2020. Spice: Self-supervised pitch estimation. In *IEEE Transaction on Audio, Speech, and Language Processing*. IEEE.

Hideki Kawahara, Masanori Morise, Toru Takahashi, Toshio Irino Ryuichi Nisimura, and Hideki Banno. 2008. Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation. In *2008 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. IEEE.

Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. 2018. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. IEEE.

Tomi Kinnunen and Rosa Gonzalez Hautamaki. 2005. Long-term f0 modeling for text-independent speaker recognition. In *Tenth International Conference on Speech and Computers*.

Yuko Kinoshita, Shunichi Ishihara, and Phil Rose. 2008. Beyond the long-term mean: Exploring the potential of f0 distribution parameters in traditional forensic speaker recognition. In *Odyssey 2008: The Speaker and Language Recognition Workshop*. ISCA.

Alexander Lerch. list of mir datasets. `https://gist.github.com/alexanderlerch/e3516bffc08ea77b429c419051ab793a`. Accessed: 2024-09-25.

Yisi Liu, Peter Wu, Alan Black, and Gopala Anumanchipalli. 2023. A fast and accurate pitch estimation algorithm based on the pseudo wigner-ville distribution. In *2023 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. IEEE.

Deborah Loakes. 2006. Variation in long-term fundamental frequency: Measurements from vocalic segments in twins' speech. In *Proceedings of the 11th Australian International Conference on Speech Science and Technology*. Australian Speech Science and Technology Association Inc.

Volodymyr Lotysh and and Pavlo Humeniuk Larysa Gumeniuk. 2023. Comparison of the effectiveness of time series analysis methods: Sma, wma, ema, ewma, and kalman filter for data analysis. In *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*.

Matthias Mauch and Simon Dixon. 2014. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. IEEE.

Max Morrison, Caedon Hsieh, Nathan Pruyne, and Bryan Pardo. 2023. Cross-domain neural pitch and periodicity estimation. In *IEEE Transaction on Audio, Speech, and Language Processing*. IEEE.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Franz Pernkopf. Ptdb-tug: Pitch tracking database from graz university of technology. `https://www.spsc.tugraz.at/databases-and-tools/`. Accessed: 2024-09-25.

Srikanth Ronanki, Oliver Watts, Simon King, and Gustav Eje Henter. 2016. Median-based generation of synthetic speech durations using a non-parametric approach. In *IEEE Workshop on Spoken Language Technology (SLT 2016)*. IEEE.

Justin Salamon, Rachel Bittner, Jordi Bonada, Juan Jose Bosch, Emilia Gómez, and Juan Pablo Bello. Synth datasets.

David Talkin. 1995. A robust algorithm for pitch tracking (rapt). In *Speech Coding and Synthesis*. Elsevier.

Nazif Can Tamer, Yigitcan Ozer, Meinard Muller, and Xavier Serra. 2023. Tape: An end-to-end timbre-aware pitch estimator. In *2023 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. IEEE.

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. 2017. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen. 2023. Rmvpe:arobustmodel for vocal pitch estimation in polyphonic music. In *Interspeech 2023*. IEEE.

A Appendix

A.1 Manual pitch annotation comparison

Two annotators manually estimated the pitch of 10 random samples from each of 10 speakers. The results, presented in Table 2 and Table 3, show a pitch difference of 5 – 20 Hz due to varying annotation methods and sample selection. Annotator 1 chose all samples from a single chapter, while Annotator 2 selected samples from multiple chapters. This variation in pitch is likely influenced by changes in recording settings, chapter content, and background noise. Despite these differences, the high Pearson correlation values indicate strong similarity between the two annotation sets. Moreover, the speaker rankings based on pitch remained consistent for most speakers, with only one exception.

Table 2: A1: Manual annotation for the 10 speakers using 10 utterances each

spk_id	Mean F0	Median F0	Std_dev
84	185.21	184.80	3.08
174	149.59	149.00	2.64
422	115.60	115.20	1.58
1993	204.87	204.70	2.70
3000	85.57	85.37	3.31
3081	237.26	222.50	23.59
6295	95.52	93.32	4.65
6319	194.90	194.70	10.89
8297	105.63	107.20	2.59
8842	174.22	174.80	2.08

Table 3: A2: Manual annotation for the 10 speakers using 10 utterances each

spk_id	Mean F0	Median F0	Std_dev
84	191.08	184.80	17.21
174	131.19	131.05	15.66
422	134.12	135.86	7.24
1993	217.61	219.30	10.11
3000	110.98	105.20	12.65
3081	230.15	230.55	18.01
6295	113.52	112.20	10.17
6319	189.77	189.90	14.76
8297	114.16	114.35	6.44
8842	192.36	187.30	20.15

The Pearson correlation coefficients for the Mean and Median F0 annotations are R=0.9672, and R=0.9693 respectively. The Spearman rank correlation for the Mean and Median F0 annotations are R=0.93939, and R=0.97576 respectively. These results indicate high correlation and high reliability between the two manual annotations.

A.2 Mean Absolute Deviation difference values

Table 4: MAD difference values for the F0 estimators

speaker_id	crepe	fcnf0	pYin	reaper	spice	swipe
84	8.15	-3.07	44.49	0.22	-1.98	-5.25
174	40.30	-1.08	47.23	-2.23	0.46	16.15
422	26.31	-0.64	23.60	-0.89	-0.19	5.65
1993	25.29	-3.77	29.05	0.00	2.78	9.91
3000	29.57	4.49	35.49	-0.66	0.09	10.62
3081	4.76	-20.32	-23.72	1.96	-0.43	-27.60
6295	27.06	2.31	37.46	-0.19	1.36	15.55
6319	27.06	-15.48	9.28	-0.68	0.17	-9.45
8297	34.52	1.94	32.74	1.35	6.40	11.42
8842	33.04	-2.38	46.74	1.46	8.57	17.87
Average	**25.61**	-3.80	**28.24**	**0.03**	**1.72**	**4.49**

Table 4 shows the computed MAD difference (with vs without post-processing) across the 10 speakers. Positive values indicate improvement (reduced variability) while negative values indicate the opposite.

A.3 Mean Absolute Error difference values

Table 5: Accuracy improvement for the F0 estimators

	mean	median (CPP)
crepe	138.26	10.52
fcnf0	30.92	25.29
pYin	114.82	53.97
reaper	8.75	10.52
spice	25.11	13.59
swipe	55.47	11.58

Table 5 shows the computed average MAE across the 10 speakers using the simple mean F0 estimation (column 1) vs post-processing/median (CPP) (column 2). These are the actual values from Figure 3.

A.4 Inter-gender rankings for 40 speakers

Expanding on Section 4.2, we estimated the pitch for all 40 speakers in the *dev-clean* subset of LibriSpeech using the CREPE algorithm, both with and without post-processing. As illustrated in Figures 5 and 6, similar to Figure 4, there was a significant improvement in speaker ranking based on mean pitch. Consistent with our findings, all male speakers exhibited lower pitch than female speakers. However, speaker 7976 displayed an unusually high pitch compared to other male speakers, potentially due to gender preference or mislabeling.

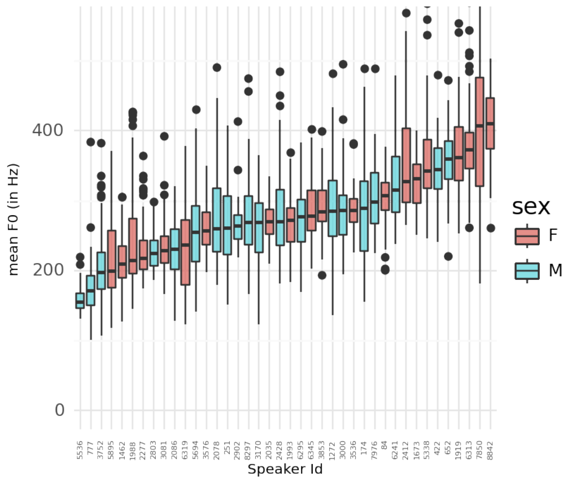


Figure 5: Inter-gender rankings for 40 speakers in LibriSpeech *dev-clean* without post-processing (using mean)

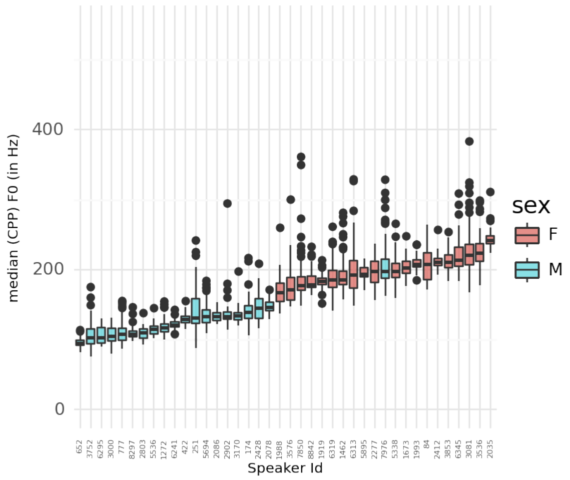


Figure 6: Inter-gender rankings for 40 speakers in LibriSpeech *dev-clean* with post-processing (*median (CPP)*

A.5 Histogram of Threshold values

Figure 7 shows the histogram plots of the confidence scores. The plots show bimodal distribution for CREPE, FCNF0, REAPER, and SPICE. Having thresholds between the peaks would be intuitive for optimal performance of each algorithm. However, for pYin and SWIPE, a continuous trend distribution does not support a justifiable robust threshold selection.

A.6 MAE vs Threshold values

Parameter sweep was done to check how the MAE varies across different thresholds. The succeeding figures show the experimental results from sweeping across a) the 10 speakers, b) only the 5 male speakers and c) only the 5 female speakers.

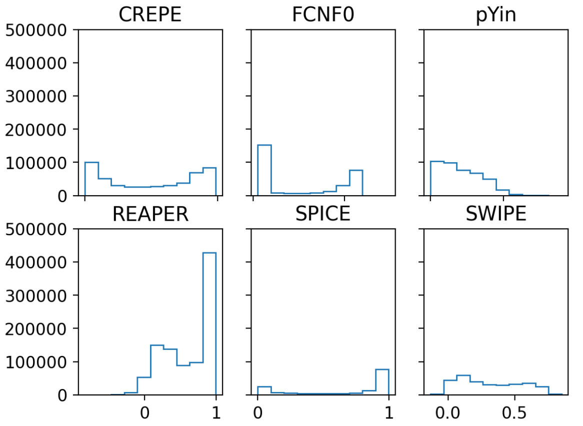


Figure 7: Histogram plot of values for confidence scores (pYIN, CREPE, SPICE) strength (SWIPE), correlation (REAPER), and periodicity (FCNF0)

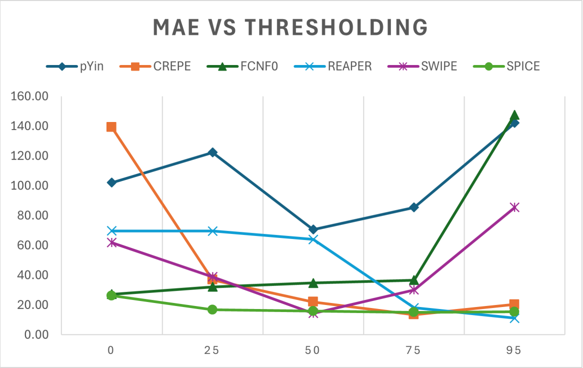


Figure 8: Mean Absolute Error vs Threshold values for all speakers

Based on Figure 8, optimal threshold values are: 0.5 for pYin and SWIPE, 0.75 for CREPE and SPICE, 0 for FCNF0 and 0.95 for REAPER.

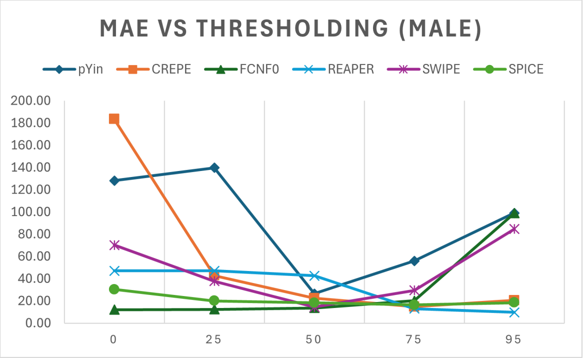


Figure 9: Mean Absolute Error vs Threshold values for male speakers

Same trend can be seen in Figure 9 with just only the male speakers.

Using only female speakers in Figure 10, optimal threshold value for pYin is now at 0 while 0.95 for

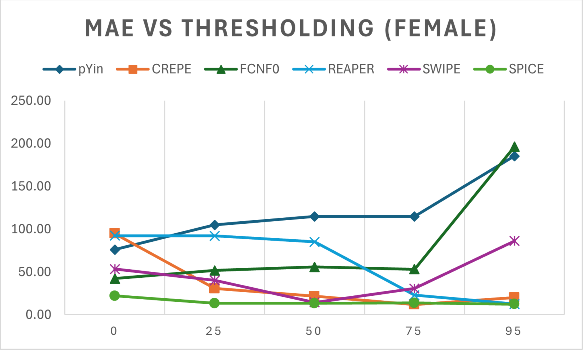


Figure 10: Mean Absolute Error vs Threshold values for female speakers

SPICE.

From the experiments on the threshold values, we observe that applying threshold for pYin will not be robust when used with a different set of data. On the other hand, we can see robust thresholding performance on CREPE, and REAPER.

A.7 Post-processing effects on Spearman Rank correlation

Table 6: Spearman Rank coefficient vs central measures

	mean	base value	median	median (CPP)
crepe-reaper	-0.28	-0.03	0.87	0.99
crepe-spice	0.05	0.33	0.93	0.99
crepe-pyin	0.56	0.77	0.96	-0.36
crepe-swipe	0.65	0.83	0.84	1.00
crepe-fcnf0	0.24	0.71	0.87	0.81

Another experiment was done to see how using the post-processed median, the mean, the median and the base value (Arantes et al., 2017) affects the rankings across the different algorithms. Looking at Table 6, we can see that rankings between CREPE, REAPER, SPICE, and SWIPE become highly correlated after post-processing.

Text-to-Speech in Azerbaijani Language via Transfer Learning in a Low Resource Environment

Dzhavidan Zeinalov[1], **Bugra Sen**[2], **Firangiz Aslanova**[2]
[1]Digital Research Lab, Baku, Azerbaijan
[2]Kapital Bank, Baku, Azerbaijan
`javidan.zeynalov@researchlab.digital`
`{bugra.sen, firangiz.aslanova}@kapitalbank.az`

Abstract

Most Text-to-Speech models cannot operate well in low-resource languages, and require a great amount of high-quality training data to be considered good enough. Yet, with the improvements made in ASR systems, it is now much easier than ever to collect data for the design of custom Text-to-Speech models. In this paper, our work on using ASR model to collect data to build a viable Text-to-Speech system for one of the leading financial institutions of Azerbaijan will be outlined. NVIDIA's implementation of the Tacotron 2 model was utilized along with the HiFiGAN vocoder. As for the training, the model was first trained with high-quality audio data collected from the Internet, then fine-tuned on the bank's single-speaker Call Center data. The results were then evaluated by 50 different listeners and got a Mean Opinion Score of 4.17, displaying that our method is indeed viable. With this, we have successfully designed the first Text-to-Speech model in Azerbaijani, and publicly shared 12 hours of audiobook data for everyone to use.

1 Introduction

Text-to-speech systems are generally made up of two parts to gain more control over the whole process: mel – spectrogram generator, which learns based on our labeled audio data to synthesize mel – spectrogram of input text, and a neural vocoder which is essentially what turns mel – spectrograms into a waveform (Shen et al., 2018). One of the most successful TTS systems, namely Tacotron 2, was released in 2016, with a performance that rivals that of professional speakers. Tacotron 2 is a system that first maps character embeddings to mel-scale spectrograms, and then utilizes a vocoder to generate audio waveforms from the spectrograms (Shen et al., 2018). By using the LJSpeech dataset that contains a single speaker data of around 24 hours, Tacotron 2 achieved an incredible performance of 4.53 MOS, almost the same score that would be given to a recording of a professional voice actor (Ito and Johnson, 2017). While TTS systems for popular languages such as English have existed for quite some time, many low–resource languages struggle in this regard. One interesting recent development is the utilization of Speech Recognition models to collect data. One of the most popular models, Whisper, which is an open-source speech recognition model released in 2022, performs extremely well in numerous languages (Radford et al., 2022). Its largest version currently supports over 100 languages, and it can be run in a Google Colab environment, making it quite accessible to users. Azerbaijani language, also known as Azeri, is a Turkic language spoken primarily in Azerbaijan. It is also spoken by many across other countries, mainly Turkey, Iran, Georgia, and Russia. There is currently little work being done about the data collection of Azerbaijani speech, making the development of Speech models from scratch impossible. Even Whisper's largest version offers only around 24.8 WER percentage, which does not make it into the top 30 (Radford et al., 2022). One interesting work we have come accross is (Kamil Aida-Zade, 2010) , which uses a simple TTS architecture. However, it is not up to date as the paper has been around for years, and there are many new TTS models that would outperform the rather statistical and probabilistic approach used by them.

In this work, our task is to develop a TTS solution for one of the leading banks of Azerbaijan, with naturalness being our chief goal. We also show the effectiveness of using 2 pre-trained models for the complete training to overcome Tacotron 2's need for large amounts of data.

2 Approach

We developed Azerbaijani TTS by using Tacotron 2 architecture together with the HiFiGAN vocoder model. The HiFiGAN model was not trained by us, and instead, the universal version – which was trained on the LJSpeech dataset – was used for inference (Kong et al., 2020).

2.1 TTS Model Architecture

The reason for choosing Tacotron 2 model was simple: our model will later be used for a plethora of possible utilizations, in each of which the voice naturalness, rather than inference speed, is crucial. Our choice of implementation was that of NVIDIA's, which is PyTorch implementation of Tacotron 2, providing faster-than-realtime inference as a nice bonus. The Tacotron 2 model itself has an encoder-decoder architecture, with location-sensitive attention being utilized. Audio data, accompanied by its transcriptions, is needed for the training of Tacotron 2. In the overall flow of the said model, text data first follows a few steps of preprocessing, namely normalization, removal of punctuations, and conversion of numerics into words. The audio data experiences the same, as the Tacotron 2 model is coded to handle audio with specific parameters, namely 22,05 kHz, 16-bit format, mono encoding, and wav file type (Shen et al., 2018). The above preprocessing was done for all the data utilized in our work before doing any training. The encoder part is responsible for turning text – a sequence of characters, to be precise – into embeddings. This enables our models to understand our data, as text data by itself cannot be processed, and embedding gives our model the semantic meaning of the text data. Then, 3 Convolutional Layers, followed by a Bidirectional LSTM layer capture the long-term dependencies within the text. This step allows our model to extract features that will later be relevant to the mel -spectrogram generation. The attention mechanism utilized in the implementation is location-sensitive attention. The mechanism allows the model to virtually direct its "attention" on the text sequence's specific parts when doing the predictions of mel – spectrogram frames (Zhang et al., 2021). There are also pre–net, and post–net layers, which are responsible for enhancing feature extraction from text and quality of synthesized mel – spectrograms respectively. Last but not least, the generated mel – spectrogram is then provided as the input to the HiFiGAN model, which generates the audio waveforms. Little to no changes were made to the model architectures for both Tacotron 2 and HiFiGAN, as they both demonstrate outstanding results on their own (Shen et al., 2018). Figure 1 display the overall flow of our development.

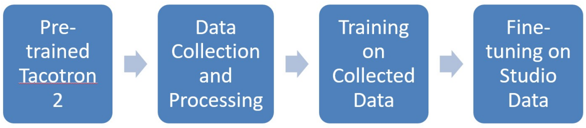

Pre-trained Tacotron 2
Data Collection and Processing
Training on Collected Data
Fine-tuning on Studio Data

Figure 1: Overall Process Flow of Azerbaijani TTS Development.

The parts that were changed are as follows:

- The letters variable in the symbols.py was changed to accommodate the Azerbaijani alphabet.
- The chosen cleaner was changed to a basic cleaner and adapted to the Azerbaijani language – specifically, handling abbreviations and numbers.
- Hyperparameters were changed as both the amount and type of data differ from the original implementation.

2.2 Data Collection and Preprocessing

In this sub-section, the data utilized, its collection as well as preprocessing will be outlined. Our first thoughts were to utilize available datasets such as Common Voice by Mozilla Foundation, or FLEURS (Ardila et al., 2020; Conneau et al., 2022). However, the data quality across many audios was too low, and as the location-sensitive attention is sensitive to the quality of training data, the idea was rejected (Zhang et al., 2021). The financial institution provided us with audio recordings that are currently utilized in its Call Center. The audios were of studio quality. In our experiments, we found the data to be not enough to capture many phonetic features of the language and therefore collected additional data. We found open-sourced audiobook recordings, which totaled 11 hours. The recordings were high quality but did not have any transcriptions, as the audiobook was based on a really old PDF edition. Hence why the transcriptions of the audio recordings for both studio data and audiobook were obtained by using OpenAI's Whisper model's large version 3 (Radford et al., 2022). By changing decoding options as well as making use of a Voice Activity Detection filter, namely Silero - VAD, we achieved accurate timestamping of the recording along with its corresponding transcriptions (Team, 2021). The decoding parameters that were changed are outlined in Table 1. As the dataset utilized for the original Tacotron 2 training was between the length of 2 to 20 seconds, we followed the same rule when segmenting our data (Shen et al., 2018). In case any audio segments were longer than the aforementioned value, it was split into parts manually, as there was a limited number of them after splitting via Whisper model (audiobook).

Table 1: Decoding Parameters of Whisper Model

Parameter Name	Value chosen
Beam size	5
Best of	5
Temperature	$(0.0, 0.2, 0.4, 0.6, 0.8, 1.0)$
Vad filter	silero:v3.1

The information regarding our data is provided as follows in Table 2.

Table 2: Audio Source, Amount, and the corresponding quality

Source	Amount(hr)	Quality	Segments
Call Center	0.76	Studio Level	532
Audiobook	11.3	High	7723

3 Experiments

In all our experiments, we divided 95 percent of our data to be training set, and the rest to be validation set. At first, only 46 minutes of Call Center audio data was available. We first conducted a fast trial by excluding any audio that was longer than 12 seconds. This left us with a total of 30 minutes of studio-quality data. The results seemed to have overfit, as some letters that were present in our training set would sometimes be mispronounced, or skipped entirely – this model will be referred to as the First Model. For this reason, we manually split the rest of the data, giving us a total of 46 minutes of audio data entirely – this model will be referred to as the Second Model. This time, some degree of hyperparameter tuning was also conducted to see the effects of longer training, decay rate, different learning rates as well as batch sizes. This second model generated intelligible results, especially in cases when the text to be synthesized contained words close to our training data – bank terminology. That said, it lacked generalization, which was a crucial aspect. Hence, the search for more data began, and we later found an audiobook of about 11 hours of data. It was split into 46 parts, each being read by the same Female speaker with a quality that was considered good enough. As we already know using a pre-trained model, even if in a different language, will still produce better results, it was decided to train the model beginning from the English language checkpoint using audiobook data (Pine et al., 2022; Byambadorj et al., 2021). Then, we would fine-tune the model with our Call Center data, not only introducing audio recordings of a higher quality but also the terminology related to finance. After training the model on audiobook for around 150,000 iterations, the results were already amazing as the model could generalize as well as produce intelligible results entirely – this model will be referred to as the Third and the Final Model. The hyperparameters we chose for this were based on the original implementation, as the model might have overfitted if we used the parameters as before (Shen et al., 2018). We stopped the training at 500 epochs and used our 46-minute Call Center data to further fine-tune it for another 300 epochs entirely – this model will be referred to as the Final Model (Byambadorj et al., 2021). The hyperparameters and additional information regarding different models are provided in Table 3.

Table 3: Parameter and data changes across models

Parameters	1st Model	2nd Model	Final Model
Epochs	250	500	500
Learning Rate	$1e^{-4}$	$5e^{-4}$	$1e^{-3}$
Weight Decay	0	$1e^{-6}$	$1e^{-6}$
Beta 1	0.99	0.99	0.99
Beta 2	0.999	0.999	0.999
Batch Size	8	16	16

4 Results

The evaluation of TTS systems is still a challenge, as there is not one metric that is universally accepted. In the case of speech recognition, there are 2 prominent methods, namely Word Error Rate (WER) and Character Error Rate (CER) (Wang et al., 2003). For TTS, the only viable metric is the Mean Opinion Score (MOS) (Viswanathan and Viswanathan, 2005). To evaluate our models, we generated a total of 100 sentences, 70 sentences similar to our training data, and 30 sentences completely new. The reason for such distribution was due to the core reason for TTS development, which was to be utilized in the banking sector. 10 independent subjects rated the model samples across 5 metrics such as naturalness, overall quality, prosody, pronunciation, and intelligibility. The subjects are all native listeners, and they have all been informed about the MOS metric and how it is used to evaluate the performance of TTS models. Despite not being experts on financial domain specifically, we believe their knowledge of the language is still enough, as the model does not dive too deep into financial terms, and generates sentences known by most speakers. Then, the average score given by each subject per sample was summed up, and divided by the number of participants to evaluate a model. The sentences were unforeseen in our training data, and the overlapping words were kept to a minimum. The scores received by the trained models are given in Table 4. However, only the Second and Final Models were evaluated due to the scarcity of time and resources for evaluators. For the convenience, Second Model will be denoted with number 2, and Final Model will be denoted with number 3. To increase the readability of the table, the following abbreviations are utilized:

- Intelligibility – I.
- Naturalness – N.
- Prosody – Py.
- Quality – Q.
- Pronunciation – Pn.
- Average – Avg.

Table 4: Mean Opinion Score for each Model

Model	**N**	**Q**	**I**	**Pn**	**Py**
2	$2,45$	$2,71$	$2,42$	$2,22$	$2,32$
3	$4,12$	$4,3$	$3,98$	$3,92$	$4,23$

Our Final Model received a Mean Opinion Score of 4.17, with a confidence score of ± 0.4, rivaling some high-resource languages.

5 Discussion

Our results show that currently, even for a language that ranks 39th on the WER evaluation of the Whisper model, it is possible to collect enough data for the design and training of a high-quality TTS system (Radford et al., 2022). Tacotron 2 architecture, despite being sensitive to data quality, is more than capable of utilizing transfer learning in the same language for a different speaker to provide a high-quality mel – spectrogram generation and the HiFiGAN model does not necessarily need to be fine-tuned for effective voice synthesis (Kong et al., 2020; Pine et al., 2022).

6 Conclusion

In this paper, we outline the works done to develop a Text-to-Speech System for the Azerbaijani language for one of the leading financial institutions of the said country. The problem of not having enough data was overcome by the collection of high-quality data from the Internet, and some hyperparameter tuning as well as additional tests were carried out to see the impact on convergence and model performance. With even further development of ASR systems, it will soon be possible to train TTS models for languages that are low-resourced. Additionally, we would like to next time set up a phoneme-dictionary-based training, which is said to improve convergence speed even further.

Limitations

While we do believe the work we have done could be helpful to others who are also trying to use ASR models for data collection purposes for Text-to-Speech applications, there is a limitation to this. We have noticed that the quality of audio largely depends on the speaker's prosody as well as the quality of audio. That is to say, to get a natural voice, it is also needed that the data collected is not monotonous, but rather rich in sounds. The Audiobook-only model that we tried was average in quality, but there was a huge difference between this model and the model that was trained on top of the audiobook with only 46 minutes of high-quality data. It is possible that a model that was trained with only 2 hours of studio quality data could surpass that of 15 hours of average quality data. In summary, if there is no high-quality data available on the Internet, the quality of TTS model might still be lacking, even if manual corrections are made to the labels.

Ethics Statement

While we are excited with the improvements made in ASR technology fields, it is crucial that the data collected is done with consent, or with data that is openly sourced. We have obtained our data from a local public library that belongs to the government with their consent. Collection of audio data and building a TTS model on someone's voice without their knowledge or consent is something we discourage strongly. With power, comes great responsibility.

Acknowledgment

The authors would like to thank the financial institution for being generous and sharing its data, as well as all that had parts in the design of the said Text-to-Speech model, no matter how small of a contribution it was.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus.

audiobook. Audiobook dataset. [link].

Zolzaya Byambadorj, Ryota Nishimura, Altangerel Ayush, Kengo Ohta, and Norihide Kitaoka. 2021. Multi-speaker tts system for low-resource language using cross-lingual transfer learning and data augmentation. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 849–853.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. Fleurs: Few-shot learning evaluation of universal representations of speech.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Aida Sharifova Kamil Aida-Zade. 2010. Azerbaijan text-to-speech synthesis system. In *The Third International Conference "Problems of Cybernetics and Informatics"*, pages 33–40.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis.

Aidan Pine, Dan Wells, Nathan Brinklow, Patrick Littell, and Korin Richmond. 2022. Requirements and motivations of low-resource speech synthesis for language revitalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7346–7359, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783.

Silero Team. 2021. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. `https://github.com/snakers4/silero-vad`.

Mahesh Viswanathan and Madhubalan Viswanathan. 2005. Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (mos) scale. *Computer Speech Language*, 19(1):55–83.

Ye-Yi Wang, A. Acero, and C. Chelba. 2003. Is word error rate a good indicator for spoken language understanding accuracy. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, pages 577–582.

Xiangzhou Zhang, Bingpeng Ma, Hong Chang, Shiguang Shan, and Xilin Chen. 2021. Location sensitive network for human instance segmentation. *IEEE Transactions on Image Processing*, 30:7649–7662.

SG-RAG: Multi-Hop Question Answering With Large Language Models Through Knowledge Graphs

Ahmmad O. M. Saleh
Sabanci University
ahmmad@sabanciuniv.edu

Gokhan Tur
University of Illinois
Urbana-Champaign
gokhan@illinois.edu

Yucel Saygin
Sabanci University
ysaygin@sabanciuniv.edu

Abstract

Large Language Models (LLM) such as GPT3 and Llama tend to hallucinate, especially for domain-specific questions. To alleviate this problem, Retrieval Augmented Generation (RAG) has been proposed but LLMs still suffer in multihop question answering even with RAG. Knowledge Graphs represent domain information in a structured manner and they have been used for reasoning in AI. In this work, we propose SubGraph Retrieval Augmented Generation (SG-RAG), a novel zero-shot Graph RAG method that exploits the structured information in Knowledge Graphs in order to accurately answer multihop questions with LLMs. We form a Cypher query based on the given question to retrieve the set of relevant subgraphs that is further provided as context to the Language Model. We implemented and tested our methodology on a benchmark question-answering data set on movies domain. Experiments show that the accuracy of 2-hop and 3-hop questions issued to LLAMA 8B Instruct and GPT4-Turbo significantly increases compared to LLAMA and GPT with and without RAG.

1 Introduction

Language Models have revolutionized how we represent knowledge and significantly impacted question-answering systems. Large Language Models (LLM) have proven to be very effective in generating convincing answers, especially for generic questions Touvron et al. (2023). However, they also tend to hallucinate when they encounter domain-specific questions Tonmoy et al. (2024). In the case of LLMs such as LLAMA, hallucination becomes a severe problem Li et al. (2024). In Table 1, we provide sample questions submitted to LLAMA3 8B Instruct where the answers show hallucinations of the model. In order to alleviate this problem, Retrieval Augmented Generation (RAG) was proposed by Lewis et al. (2020). With RAG, questions are answered based on a set of documents where documents most similar to the given question are retrieved and provided as context to the LLM. The semantic similarity of a question to the documents is calculated through word embeddings and the top few documents are provided as context. RAG eliminates most of the hallucinations in the case of single-hop questions such as "When has been the release year of the film No Looking Back", but for multihop questions like "Senator William Broyles Jr. wrote films with whom" (2-hop) and "When were the release years of the films led by Edges of the Lord as director"(3-hop), LLAMA3 8B Instruct fails to give correct answers. In order to understand the degree of hallucination, we evaluated LLAMA 8B on a benchmark Question-Answer data set where the questions and corresponding answers are provided. We observed that single-hop questions are answered with high accuracy, while for 2-hop questions the accuracy drops drastically, and for 3-hop questions the accuracy decreases even further.

In order to improve their performance, an alternative form of giving context to LLMs was proposed in the form of Knowledge Graphs (KGs). KGs provide domain information in a structured way. The term Graph RAG was coined in a blog by Microsoft Research Larson and Truitt (2024) where the authors highlighted the limitations of the standard RAG method in answering questions that require multiple pieces of information. They suggested transforming the unstructured documents into a knowledge graph as a solution to those limitations.

In this work, we introduce the SubGraph Retrieval Augmented generation (SG-RAG), a novel zero-shot Graph RAG method that exploits the relations stored in KGs to answer questions. An overview of SG-RAG is demonstrated in Figure 1. SG-RAG uses Cypher statements representing the semantics of the questions to retrieve the set of subgraphs containing relevant information from KG. SG-RAG then transforms the subgraphs into a tex-

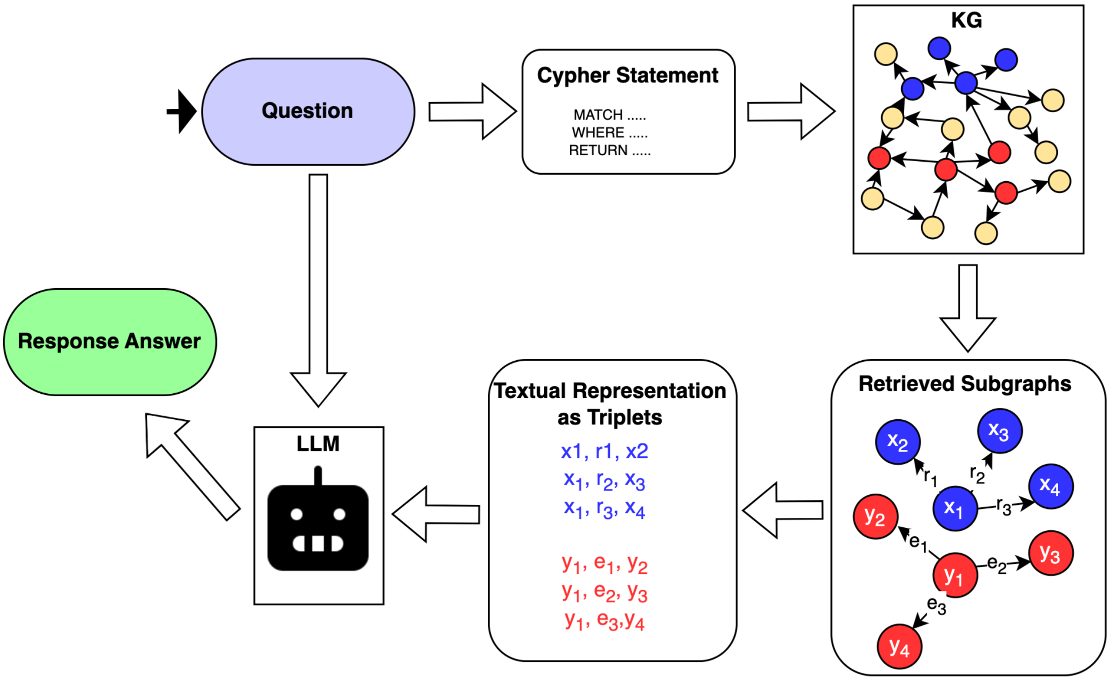


Figure 1: An Overview of SG-RAG Methodology.

tual representation in the form of triplets. Triplets are partitioned into groups based on the retrieved subgraphs as highlighted in Figure 1. Triplets are finally provided to the LLM as context to generate an answer. The task instruction sent to LLM highlights the structure of the triplets by using $(subject, relation, object)$ format where the direction of the $relation$ from $subject$ to $object$ is emphasized. The experiments show that SG-RAG provides more accurate answers compared to LLM with and without RAG, especially for multihop questions.

2 Background and Related Work

Large Language models (LLMs) are being used for language understanding and question answering Touvron et al. (2023); Reid et al. (2024). However, LLMs struggle to provide accurate answers for domain-specific questions, generating factually wrong, or irrelevant random responses which are referred to as model hallucination Tonmoy et al. (2024). Retrieval Augmented generation (RAG) works on solving the hallucination problem by embedding chunks of the domain information in the LLM prompt with the question Lewis et al. (2020); Gao et al. (2023). The choice of the information chunks shared with LLM is determined by measuring the semantic similarity between the question and the domain knowledge information. The RAG method with LLMs has been applied to question-answering systems targeting a wide range of domains including Finance by Setty et al. (2024), Medicine by Zakka et al. (2024), and Religion by Alan et al. (2024) to name a few. RAG method with semantic similarity search decreases the LLMs' hallucination on simple domain-specific questions, however, the semantic similarity search is not able to retrieve the indirect information required to answer multihop questions Larson and Truitt (2024).

After the success of LLMs on many Natural Language Processing tasks, researchers started to investigate the potential of using LLMs on graphs Jin et al. (2023). such as using LLMs on classifying graph nodes Chen et al. (2024). Edge et al. (2024) use LLMs to convert unstructured knowledge into a knowledge graph to answer global questions that require a global awareness of the domain knowledge. The method proposed by Edge et al. (2024) is based on creating groups, "communities" as they call them, of subgraphs. Each community has a summarization of the information stored in the community subgraph. The answer to any global question is based on collecting the summarization from the communities and providing an answer based on them. The most relevant work to ours is a recent paper by Jin et al. (2024) where authors propose Graph Chain-of-Thought (Graph-COT) approach. Their aim is to tackle questions that require reasoning by giving LLM the ability to interact with the knowledge graph directly through

Table 1: Examples of the LLAMA3 8B Instruct hallucination on multihop questions from movies domain (with and without RAG).

1-hop	Question	When has been the release year of the film No Looking Back
	Gold Answer	1998
	LLAMA3-8B	2019
	LLAMA3-8B with RAG	June 28, 2024 | Japanese theaters | Annecy International Animation Film Festival (June 9-15, 2024) | Annecy Presents category | Annecy International Animation Film Festival | June 28, 2024 | Japanese theaters | Annecy International Animation Film Festival
2-hop	Question	Senator William Broyles Jr. wrote films with whom
	Gold Answer	Alvin Sargent, Claude Chabrol
	LLAMA3-8B	Tom Cruise Explanation: Tom Cruise and William Broyles Jr. co-wrote the film "The Polar Express". William Broyles Jr. is a screenwriter and his films include "The Polar Express" and "Cast Away". Tom Cruise is an actor who has starred in several films, including "Top Gun"
	LLAMA3-8B with RAG	Robert Redford, Steven Spielberg, Tom Cruise, Barry Levinson, Oliver Stone, Ron Howard, James Cameron, Brian De Palma, Martin Scorsese, Robert Zemeckis, Barry Sonnenfeld, Edward Zwick, Steven Soderbergh, Clint Eastwood, Sam Raimi
3-hop	Question	When were the release years of the films led by Edges of the Lord as director
	Gold Answer	1987, 1993
	LLAMA3-8B	1. The Lord of the Rings: The Fellowship of the Ring (2001) - December 19, 2001 2. The Lord of the Rings: The Two Towers (2002) - December 18, 2002 3. The Lord of the Rings: The Return of the King (200
	LLAMA3-8B with RAG	2001, 2002, 2005, 2007, 2008, 2002, 2005

a set of predefined functions. Hence, the LLM has the freedom to decide when and how to traverse the graph to gain the information needed to answer the question. The graph description, including the nodes' type, attributes, and outer edges, is augmented in the LLM prompt. The main limitation of Graph-COT is that the model can reach a dead-end in cases where the initial node does not have any outer edge as demonstrated in Figure 3 where the question is asking about the release years of the movies *Sharon Tate* acted. In this case the LLM in Graph COT will start with *Sharon Tate* node. Since all the edges connected to the *Sharon Tate* node are incoming edges as in Figure 2, LLM will not be able to traverse other nodes to find the release years of *The Wrecking Crew* and *Valley of the Dolls* from *Sharon Tate*, therefore LLM will not be able to answer the question correctly. Another point is that Graph-COT works on GPT3.5 Turbo which is an advanced black box model, however, when we run Graph-COT on LLAMA3-8B which is an open-source model with a much lower number of parameters compared to GPT, we observed many hallucinations for our benchmark questions.

Other individual and commercial experiments have been conducted on the LLM and KG, as highlighted by Kollegger (2024) stressing the importance of using KG with LLM and providing approaches to combine them, and the blog-post written byAlto (2024) as an implementation tutorial of applying a hybrid approach of RAG and KG with LLM using LangChain and Neo4j graph database. In this paper, we propose a novel Graph RAG methodology based on subgraph retrieval that we call SG-RAG to address the problem of multi-hop question answering.

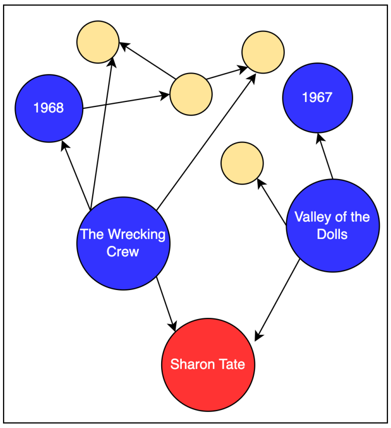
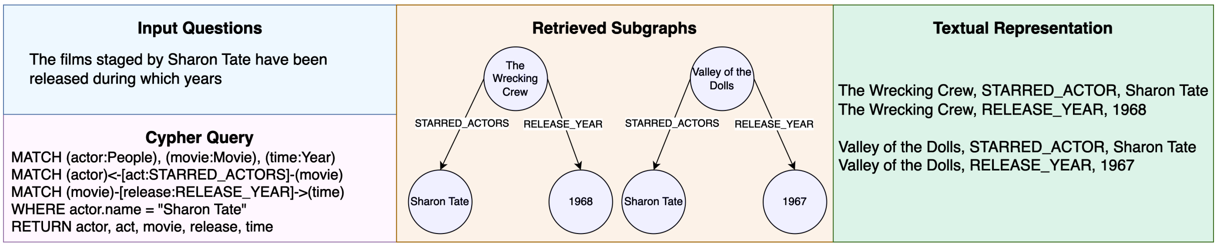


Figure 2: An Illustration of a Dead-end Scenario for Graph-COT.

3 Preliminaries and Problem Definition

In the following paragraphs, we define preliminary concepts that will be used in the problem statement.

Definition 3.1. Graph. A graph $G = (V, E)$ is a data structure consisting of a set of nodes, denoted by V, and a set of edges, denoted by E. For any



The films staged by Sharon Tate have been released during which years

```
MATCH (actor:People), (movie:Movie), (time:Year)
MATCH (actor)<-[act:STARRED_ACTORS]-(movie)
MATCH (movie)-[release:RELEASE_YEAR]->(time)
WHERE actor.name = "Sharon Tate"
RETURN actor, act, movie, release, time
```

The Wrecking Crew, STARRED_ACTOR, Sharon Tate
The Wrecking Crew, RELEASE_YEAR, 1968

Valley of the Dolls, STARRED_ACTOR, Sharon Tate
Valley of the Dolls, RELEASE_YEAR, 1967

Figure 3: An Example of the SG-RAG Retrieval for a 2-hop Question.

edge $e_i \in E$, there exists two nodes, $v_j, v_k \in V$, such that e_i connects v_j and v_k. A graph can be directed or undirected where the edges in the former have a direction.

Definition 3.2. Subgraph Given a graph $G = (V, E)$, a graph $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$.

Definition 3.3. Knowledge Graph. A knowledge Graph, KG, represents domain knowledge in a graph data structure. In KG, a node represents a unit of information and an edge represents the relation(s) between the two nodes. Nodes and edges may have a label describing the type of knowledge or relations in the nodes or edges respectively. Each node $v_j \in V$ may have extra information embedded in the form of attributes. Attributes can differ based on the type of the nodes. For example in case of Movies KG, the possible node labels are Movies, People, Genre,...etc. The set of node labels with the candidate edge types represents the schema of the KG. Table 2 describes the schema of MetaQA-KG that we used in our experimental evaluation.

Definition 3.4. n-hop Question. An n-hop question is a question that requires one or more subgraphs from a KG such that each subgraph contains n edges. For example, in order to answer the 2-hop question in Figure 3, we need two subgraphs where each subgraph contains 2 edges, "STARRED _ACTORS" and "RELEASE _YEAR".

Problem Definition: For any domain D represented by a knowledge graph KG, our aim is to accurately answer n-hop questions Q about D. We assume that the questions are about the entities in KG with specific relations to other entities. For example, for the 1-hop question: "What are the movies directed by Sharon Tate?" we are interested in the *Movies* entity that is related to *Sharon Tate* entity with *directed by* relation. The expected answers to the questions in Q are the set of entities that satisfy the constraints in the form of relations provided in the question.

4 SG-RAG Methodology for Multihop Question Answering

In this section, we present SubGraph Retrieval Augmented Generation (SG-RAG), a novel zero-shot Graph RAG method for answering domain-specific multihop questions using KG and LLM. SG-RAG has two main steps, subgraph retrieval and response generation. The Subgraph Retrieval is based on querying KG using a Cypher statement representing the input question and then transforming the retrieved subgraphs into a set of triplets. The response generation step takes the input question and the resulting triplets from the retrieval step and augments them into a prompt with an instruction to LLM, then the prompt is sent to LLM to generate a response to the question. The flow of SG-RAG is demonstrated in Figure 1. The following subsections explain the subgraph retrieval and response generation in detail.

4.1 Subgraph Retrieval

Rather than retrieving a specific piece of information from KG as in Bratanič (2024); Alto (2024) such as list of movie names, the subgraph retrieval step relies on Cypher statements to retrieve a set of subgraphs from the KG containing the required relevant information to answer the input question. Then, SG-RAG transforms the retrieved subgraphs into a textual representation that will be provided as context to the LLM prompt during the response generation step.

Querying the Knowledge Graph: Cypher is a query language design by Neo4j for property graphs built following Graph Theory Francis et al. (2018). For an input question q, we use a statement in Cypher Query Language to be executed on the knowledge graph. The Cypher statement searches for the set of subgraphs with nodes containing the answer of q, then returns the found subgraphs as records. In the example shown in Figure 3, the Cypher statement aims to retrieve the movies in which Sharon Tate acts together with the release years of those movies. The result of executing the Cypher statement is two subgraphs shown in Figure 3. We use domain-specific Cypher templates to generate Cypher queries for the benchmark questions.

Transformation of Subgraps into Textual Representation: The transformation is based on converting each pair of nodes connected by an edge into a triplet of the form $(Subject, Relation, Object)$. More Precisely, given two nodes n and m connected by a directed edge e from n to m, the resulting triplet will be (n, e, m). Textual transformation needs to preserve the partial order imposed by the retrieved subgraphs. Therefore the triplets from the same subgraph are grouped together. Grouping triplets based on the subgraphs helps the LLM extract the correct information and prevents it from getting confused between the different subgraphs. In Figure 3, two retrieved subgraphs were depicted where each subgraph contains 2 edges, hence the textual representation includes 4 triples split into two groups.

4.2 Response Generation

The LLM prompt used to generate a response contains the task instruction, the context as the set of triplets coming from the retrieval step, and the input question. The task instruction is a simple instruction explaining the task to the LLM and describing the structure of the triplets. During our initial trials, we explored different prompt templates that differ in the task instruction such as $(entity, relation, entity)$ which does not capture the directed edge structure of the KG. Using the $(subject, relation, object)$ gave the best results since it provides the LLM additional information about the direction of the relation such that the $relation$ is from $subject$ to $object$. The final version of the prompt template we used is demonstrated in Figure 4. After creating the prompt based on the prompt template, it is sent to the LLM to generate a response.

5 Experimental Setup

5.1 Dataset

MetaQA is a benchmark dataset introduced by Zhang et al. (2018). It includes a knowledge graph (MetaQA-KG) based on data about movies. In addition to the knowledge graph, it contains question-answer pairs about MetaQA-KG. The questions are generated through templates, and a paraphrased version of the questions called NTM is created by translating them to French and then back to English. Each question has a single category out of 49 categories. The question-answer pairs are divided into 1-hop, 2-hop, and 3-hops. For our experiments, we randomly picked 15K NTM questions with equal number of 1-hop, 2-hop, and 3-hop questions.

Response Generation Prompt Template

I will give you a context of information as triples of subject, relation, object.
Answer the following question Based on the given context.
Context:
{context}
Question: {question}
Answer:

Figure 4: The Prompt Template Used for SG-RAG Response Generation.

MetaQA-KG contains 9 types of relations: "directed by", "written by", "starred actors", "release year", "in language", "has tags", "has genre", "has imdb votes", and "has imdb rating". Based on the semantics of those relations, we divided the entities into 8 groups: Movies, People, Year, Language, Tag, Genre, IMDB Votes, and IMDB Rating. Hence, the Graph Schema of the MetaQA-KG becomes as shown in Table 2 such that each entity has a single attribute called name, while relations don't have attributes.

Table 2: The Graph Schema of MetaQA-KG after grouping the entities based on the semantics of the relations.

(:Movies)-[:DIRECTED _BY]->(:People)
(:Movie)-[:WRITTEN _BY]->(:People)
(:Movie)-[:STARRED _ACTORS]->(:People)
(:Movie)-[:IN _LANGUAGE]->(:Language)
(:Movie)-[:RELEASE _YEAR]->(:Year)
(:Movie)-[:HAS _GENRE]->(:Genre)
(:Movie)-[:HAS _TAGS]->(:Tag)
(:Movie)-[:HAS _IMDB _VOTES]->(:Vote)
(:Movie)-[:HAS _IMDB _RATING]->(:Rate)

5.2 Baselines

We consider the following baselines in our experiment:

- LLM: Using the LLM alone to answer the questions. The answers are based on the internal knowledge stored in the model's parameters.

- RAG: It is based on the original RAG method proposed by Lewis et al. (2020). The external knowledge is represented by a set of plain-text documents.

5.3 Implementation Settings

Since the MetaQA benchmark does not contain the Cypher queries, we generated them based on templates. The generation process is based on creating a Cypher query template for each category. A subset of the query templates is provided in Table 3. Cypher statements are generated by replacing the "<entity>" tag with the entity name in the corresponding question.

Our baseline RAG Lewis et al. (2020) is based on indexing plain-text documents into a vector database using textual embedding. Since the knowledge in MetaQA is a graph structure, we retrieved Wikipedia documents about the entities that appear in our test questions. The retrieved Wikipedia documents are split into chunks with a maximum size of 100 words as in Lewis et al. (2020) that are indexed into a vector database. We used LLAMA-3 8B Instruct version AI@Meta (2024) as the backbone LLM for the baselines and SG-RAG.

5.4 Evaluation Metric

We evaluate the performance of SG-RAG and the baselines using the answer-matching rate inspired by the notion of entity-matching rate proposed by Wen et al. (2017) to evaluate the dialogue systems. The answer matching rate measures the ratio of the gold answers contained in the generated response. More specifically, let q be an input question, $Y = y_1, y_2, .., y_m$ be the gold answer, and $Y' = y'_1, y'_2, .., y'_n$ be the generated response, then:

$$MatchingRate(q) = \frac{\mid Y \cap Y' \mid}{\mid Y \mid} \tag{1}$$

The gold answers in MetaQA are a set of entity names whereas the LLM responses have a paragraph structure with explanations. Therefore, we have decided to use matching rate metric which considers only the part of the LLM generated text that is within the scope of our knowledge base.

6 Results and Discussion

Using the MetaQA dataset and the matching rate metric, we evaluated SG-RAG and compared it with the baselines. The results are demonstrated in Table 4. From the result in Table 4, we observe that the performance of the LLM alone is poor compared to other methods. This shows that relying on LLM internal knowledge alone is not enough to answer questions on a specific domain, such as Movies.

RAG has better performance compared to the LLM alone. However, the performance of RAG decreases for 2-hop and 3-hop questions. The reason behind that is the external knowledge shared with the LLM as a context is determined by the semantics of the question which is not enough to know the extra information required to answer the question. Coming back to the example in Figure 3, using the semantics of the question, RAG retrieved the documents about "Sharon Tate" which include the names of the movies she acted such as "The Wrecking Crew" and "Valley of the Dolls", but those documents do not contain extra information about the movies such as the release year, the language, or the name of the cast. RAG cannot retrieve all the necessary documents about "The Wrecking Crew" or "Valley of the Dolls" by the mere semantics of the question. This problem of RAG is addressed by SG-RAG which we can observe in Table 4 where SG-RAG outperforms the baseline methods for 1-hop, and even more for 2-hop, and 3-hop questions. SG-RAG uses the KG as an external knowledge source where the relations between the entities are represented in the structure of the graph. Moreover, we use Cypher queries to retrieve information from the KG and fully capture the structural information provided by the KG. This can also be seen in the example provided in Figure 3 where Cypher query asked to retrieve all the movies in which "Sharon Tate" was an actress, and the release year of those movies. This way, the LLM received all the information needed to answer the question.

Generating Documents based on Knowledge Graph: The low performance of the RAG with Wikipedia documents on the 1-hop questions may be caused by the fact that Wikipedia does not include the answers to our questions. To analyze that issue, we also generated documents based on the information in our knowledge graph. The generation process started with extracting the entities in our questions. Then, for each entity, we extracted the subgraph containing the targeted entity node and the neighborhood of the node. After that, we asked Gemini 1.5 Flash to generate a 100-word document about the targeted entity containing the information in the subgraph. The subgraph is embedded in the Gemini prompts as a set of triplets. Figure 5 shows the prompt template we used to construct

Table 3: Sample question categories and their corresponding Cypher templates.

Type	Category	Cypher Template

1-hop	movie to language	MATCH (m:Movie)-[r:IN _LANGUAGE]->(l:Language)
		WHERE m.name="<entity>"
		RETURN m, r, l

	director to movie	MATCH (m:Movie)-[r:DIRECTED _BY]->(d:People)
		WHERE d.name="<entity>"
		RETURN m, r, d

2-hop	writer to movie to genre	MATCH (w:People)<-[r1:WRITTEN _BY]-(m:Movie)
		-[r2:HAS _GENRE]->(g:Genre)
		WHERE w.name="<entity>"
		RETURN w, r1, m, r2, g

	actor to movie to year	MATCH (a:People)<-[r1:STARRED _ACTORS]-(m:Movie)
		-[r2:RELEASE _YEAR]->(y:Year)
		WHERE a.name="<entity>"
		RETURN a, r1, m, r2, y

3-hop	movie to director to movie to actor	MATCH (m1:Movie)-[r1:DIRECTED _BY]->(d:People) <-[r2:DIRECTED _BY]-(m2:Movie) -[r3:STARRED _ACTORS]->(a:People) WHERE m1.name="<entity>" RETURN m1, r1, d, r2, m2, r3, a
	movie to writer to movie to language	MATCH (m1:Movie)-[r1:WRITTEN _BY]->(w:People) <-[r2:WRITTEN _BY]-(m2:Movie) -[r3:IN _LANGUAGE]->(l:Language) WHERE m1.name="<entity>" RETURN m1, r1, w, r2, m2, r3, l

Table 4: The evaluation results of SG-RAG with LLAMA3-8B Instruct and the baselines on the MetaQA selected test set.

	1-hop	2-hop	3-hop
LLAMA3-8B	0.24	0.13	0.17
RAG(Wiki Docs) Top-1	0.33	0.19	0.21
RAG(Wiki Docs) Top-2	0.36	0.20	0.20
RAG(Wiki Docs) Top-3	0.38	0.22	0.20
RAG(Wiki Docs) Top-5	0.40	0.23	0.18
RAG(Wiki Docs) Top-10	0.42	0.27	0.19
SG-RAG	0.90	0.73	0.58

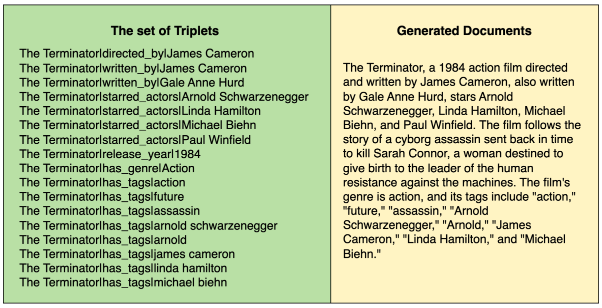
the templates we sent to Gemini to generate the document. Figure 6 provides an example of the generated document about *The Terminator* movie by Gemini based on the set of triplets representing the subgraph containing *The Terminator* node and its neighborhood. We randomly sampled a set of 1547 1-hop questions, 1589 2-hop questions, and 1513 3-hop questions, to apply this experiment within a limited time frame. From the results in Table 5, we can see that applying RAG on the generated documents based on KG achieved higher performance than the RAG on Wikipedia documents since each document contains the information of a 1-hop neighborhood around the targeted entity. However, the performance of RAG on both the generated and Wikipedia documents is comparable with 2-hop and 3-hop questions while SG-RAG has superior performance for 1-hop, 2-hop, 3-hop questions.

Document Generation Prompt Template

Write a paragraph to me about "{entity}" using these relation triplets. The paragraph should include all the information in the relation triples. Each triplet is separated by ' ; '. The paragraph should be at most 100 words long.
the relation triples:
{triplets}

Figure 5: The Prompt Template Used With Gemini for Documents Generation

Using GPT4-Turbo as backbone LLM: The low performance of RAG compared to SG-RAG even on the Gemini generated documents may be caused by the LLAMA3-8B Instruct that we chose as a backbone LLM for our evaluation. To analyze that issue further, we evaluated SG-RAG, and RAG on the Gemini generated documents on GPT4-Turbo. We did this experiment on the same small test set we used earlier to apply this experiment within a limited time frame. From the results in Table 6, we can see the superior performance of SG-RAG on 1-hop, 2-hop, and 3-hop questions. For RAG, we can notice that increasing the number of documents shared with GPT4 on 2-hop and 3-hop questions affected GPT4 negatively and decreased its performance.



The set of Triplets

The Terminator|directed_by|James Cameron
The Terminator|written_by|James Cameron
The Terminator|written_by|Gale Anne Hurd
The Terminator|starred_actors|Arnold Schwarzenegger
The Terminator|starred_actors|Linda Hamilton
The Terminator|starred_actors|Michael Biehn
The Terminator|starred_actors|Paul Winfield
The Terminator|release_year|1984
The Terminator|has_genre|Action
The Terminator|has_tags|action
The Terminator|has_tags|future
The Terminator|has_tags|assassin
The Terminator|has_tags|arnold schwarzenegger
The Terminator|has_tags|arnold
The Terminator|has_tags|james cameron
The Terminator|has_tags|linda hamilton
The Terminator|has_tags|michael biehn

Generated Documents

The Terminator, a 1984 action film directed and written by James Cameron, also written by Gale Anne Hurd, stars Arnold Schwarzenegger, Linda Hamilton, Michael Biehn, and Paul Winfield. The film follows the story of a cyborg assassin sent back in time to kill Sarah Connor, a woman destined to give birth to the leader of the human resistance against the machines. The film's genre is action, and its tags include "action," "future," "assassin," "Arnold Schwarzenegger," "Arnold," "James Cameron," "Linda Hamilton," and "Michael Biehn."

Figure 6: The set of Triplets Representing the Subgraph Containing *The Terminator* Node and Its Neighborhood on the Left and Gemini Generated Document on the Right.

Table 5: Comparison between SG-RAG, RAG on Wikipedia documents, and RAG on Gemini generated documents using LLAMA3-8B Instruct.

	1-hop	2-hop	3-hop
RAG(Wiki Docs) Top-1	0.33	0.19	0.21
RAG(Wiki Docs) Top-2	0.35	0.20	0.20
RAG(Wiki Docs) Top-3	0.36	0.22	0.20
RAG(Generated Docs) Top-1	0.64	0.15	0.17
RAG(Generated Docs) Top-2	0.66	0.12	0.13
RAG(Generated Docs) Top-3	0.66	0.12	0.16
SG-RAG	0.91	0.72	0.60

7 Conclusions

LLM with RAG has significantly impacted question-answering systems in multiple domains such as Finance by Setty et al. (2024), Medicine by Zakka et al. (2024), and Religion by Alan et al. (2024), to name a few. However, RAG is still suffering from hallucinations on multi-hop questions. In this work, we propose SG-RAG, a zero-shot Graph RAG method to answer multi-hop domain-specific questions that use Cypher statement representing the question to retrieve the set of subgraphs containing the required information to answer the question. SG-RAG is a method designed to exploit the structured information in Knowledge Graphs to increase the LLMs performance on multi-hop domain-specific questions. For an input question, SG-RAG uses a Cypher query representing the input question to retrieve the set of subgraphs containing the required information, then shares it as a context to the LLM. We evaluate our method on a question-answering benchmark dataset on movies. Our experiments show a significant increase in performance in general and specifically on 2-hop and 3-hop questions.

Table 6: Comparison between SG-RAG, and RAG on Gemini generated documents using GPT4-Turbo.

	1-hop	2-hop	3-hop
RAG(Generated Docs) Top-1	0.765	0.286	0.204
RAG(Generated Docs) Top-2	0.776	0.181	0.177
RAG(Generated Docs) Top-3	0.784	0.179	0.180
SG-RAG	0.941	0.815	0.520

Limitations

This work mainly focuses on introducing SG-RAG as a zero-shot Graph RAG method to answer multi-hop domain-specific questions. During our experiment, the Cypher statements are generated manually using domain-specific Cypher templates. To overcome the challenge of manually generating the domain-specific Cypher templates, we are working on automatically generating the Cypher statement representing the targeted question based on the KG schema as an extension to SG-RAG. In our initial trials, we observed that LLAMA3-8B and Gemini are very poor at generating valid Cypher queries. GPT-4 can generate Cypher queries, but accuracy needs to be improved. In order to address this problem we plan to fine-tune an LLM such as LLAMA3-8B to give it the ability to generate a Cypher query given the question and the graph schema.

Within the limited time frame, we evaluated SG-RAG on GPT4-Turbo over a small test set; however, we are working on extending the evaluation over a larger sample size and comparing its performance with Graph COT proposed by Jin et al. (2024).

Ethics Statement

Large Language Models (LLM) have achieved outstanding performance in natural language processing and generation, specifically in question-answering systems Touvron et al. (2023). However, the hallucination of these models can generate factual mistakes in answers or misleading information Tonmoy et al. (2024) that can be later propagated amoung people as facts. We are proposing SG-RAG as a potential solution to reduce and eradicate misinformation by exploiting the structured information in Knowledge Graphs to increase the LLMs performance on multi-hop domain-specific questions.

Acknowledgements

This research was partially funded by The Scientific and Technological Research Council of Turkey (TUBITAK)'s Industrial PhD program under project number 118C056.

References

AI@Meta. 2024. Llama 3 model card.

Ahmet Yusuf Alan, Enis Karaarslan, and Omer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint arXiv:2401.15378*.

Valentina Alto. 2024. Introducing graphrag with langchain and neo4j. Accessed 25/06/2024.

Tomaž Bratanič. 2024. Using a knowledge graph to implement a rag application. Accessed 25/06/2024.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 international conference on management of data*, pages 1433–1445.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Andreas Kollegger. 2024. Knowledge graphs for rag. `https://www.deeplearning.ai/short-courses/knowledge-graphs-rag/`. Accessed 27/06/2024.

Jonathan Larson and Steven Truitt. 2024. Graphrag: Unlocking llm discovery on narrative private data. Accessed 25/06/2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Spurthi Setty, Katherine Jijo, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *arXiv preprint arXiv:2404.07221*.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. 2024. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.

Linking Quran and Hadith Topics in an Ontology using Word Embeddings and Cellfie Plugin

Ibtisam Khalaf Alshammari
University of Hafr Al Batin,
Hafr Al Batin, Saudi Arabia
University of Leeds, Leeds, UK
`ml18ikfa@leeds.ac.uk`

Eric Atwell
University of Leeds,
Leeds, UK
`e.s.atwell@leeds.ac.uk`

Mohammad Ammar Alsalka
University of Leeds,
Leeds, UK
`m.a.alsalka@leeds.ac.uk`

Abstract

Qur'an and Hadith are the sacred texts of the Islamic religion. Arabic Qur'an and Hadith texts have been analyzed and annotated by researchers using a variety of domains, representations, and formats to improve the accessibility of Islamic knowledge. However, the many and diverse Islamic resources raise a potential challenge in linking and integrating them. The main objective of this work is to link Qur'an and Hadith topics and integrate them with related knowledge from different Islamic resources. The proposed methodology is to use a combination of word embeddings-based BERT with the Cellfie tool to achieve more accurate and meaningful data integration. The results of using the CL-AraBERT word embedding model display efficiency performance in F1 score and accuracy metrics with 91% and 84% respectively. At the same time, the constructed ontology, RQHT, links the Qur'an and Hadith topics with their related knowledge properly and consistently.

1 Introduction

Islamic research has contributed to the generation of many different databases from various sources, including the Qur'an and Hadith, which may present opportunities and challenges for religious researchers. The Qur'an and Hadith are the sacred texts of Islamic law and are considered significant historical documents that illustrate the origins of the Islamic faith. Integrating the diverse Islamic datasets can be essential for deriving comprehensive insights and enabling data-driven decision-making, thus easily leveraging them in future studies. However, the heterogeneous nature of Islamic knowledge, involving data structure, domains, and formats, could impede effective data combining.

Natural language processing (NLP) advancements have provided promising solutions to address such challenges in Arabic religious texts. Recently, word embeddings using Bidirectional Encoder Representations from Transformers (BERT) have become increasingly popular in achieving state-of-the-art performances on several NLP downstream tasks (Devlin et al., 2019). It can efficiently understand and process human language by generating contextual embeddings and linking related Islamic data.

In addition to advances in NLP, the adoption of ontology development has increased lately in the Islamic domain. Ontology can structure and represent knowledge explicitly in a machine-readable format that may be integrated into computer-based applications and systems (Gruber, 1995). The consistent interpretation and integration of data across different Islamic datasets can be facilitated by leveraging the Cellfie Plugin tool. Therefore, the synergy between word embeddings and Cellfie Plugin techniques holds great potential for Qur'an and Hadith topics linking and integration.

This paper is organized as follows: Section 2 reviews the previous studies and related work of word embeddings and ontology design. The detailed methodology to link Qur'an and Hadith topics is described in Section 3. Then, Section 4 provides an overview of the evaluation process. Results and discussion of the experiment are detailed in Section 5, which is followed by future work and conclusion in Section 6.

2 Related Work

2.1 Word Embeddings for Detecting Arabic Qur'an and Hadith Semantic Similarities

Several research studies have employed various word embedding techniques to enhance the findings of semantic similarity in the Arabic language, especially the classical texts of the Qur'an and Hadith. Abdelghany et al. (2020) presented a study using the unsupervised learning algorithm Doc2vec to identify Hadith similarity either in Matn or

Sanad across various Hadith books. Similarly, Alshammeri et al. (2021) provides research on using Doc2vec embeddings to capture the Qur'anic verses' semantics and classify similar documents into 15 predefined classes. The proposed model achieved 68% and 56% in classification accuracy and F1 score, respectively.

Alsaleh et al. (2021) implemented AraBERTv0.2 and AraBERTv2 language models to determine the semantic relatedness of Qur'anic verses pairs from the QurSim dataset. The AraBERTv0.2 language model obtained the best result with a 92% accuracy score.

Another study by (Alshammeri et al., 2022) combined a pre-trained AraBERT language model and Siamese transformer-based architecture to detect semantic similarity in the Qur'an text. The suggested approach was performed efficiently by achieving a 95% F1 score on the Qur'anic dataset.

2.2 Ontology Development

Many studies have focused on building Arabic ontologies covering different aspects of the Qur'an and Hadith knowledge. The popular Qur'an ontology was created by Hakkoum and Raghay (2015) is QuranOntology. Hybrid methods, such as Protégé-OWL and Jena TDB with Fuseki server, were used to develop the Qur'an ontology. A semantic-based search engine was established to support SPARQL queries in this ontology. A framework was proposed by (Alshammari et al., 2022) for linking and mapping Qur'anic ontologies. It combined RDF (resources description framework) Mapping Language (RML), Cellfie Plugin, and SDM-RDFizer to integrate the morphological annotations and syntactic analyses into Qur'an chapters, verses, and words in Quranicontology. The integrated onology was evaluated by a SPARQL server.

Al-Sanasleh and Hammo (2017) presented a prophetic ontology based on data derived from the Qur'an, Hadith, and explanatory book resources. They used the METHONTOLOGY approach, proposed by Fernández-López et al. (1997), to build their ontology from scratch. Likewise, Fairouz et al. (2020) developed an ontology based on Arabic Hadith texts using the METHONTOLOGY methodology. Their ontology's scope focused on the essential concepts mentioned in the EL-Bukhari book, especially the Knowledge related to Wudhu2, pillars of Islam, and worship.

Alsalhee and Abdullah (2022) conducted a study to construct an ontology that represents the Qur'anic stories. The object role modeling (ORM) and MappingMaster domain-specific language techniques were applied to build conceptual structure and convert Excel sheets to an OWL format. This ontology was evaluated using the SPARQL query language.

A study was conducted by (Altammami et al., 2021) to investigate the appropriateness of utilizing a Qur'an ontology as a foundation for linking Qur'an and Hadith ontology. A corpus-based evaluation approach was employed to evaluate the existing Qur'anic ontologies. The result of this study is that the QuranOntology by Hakkoum and Raghay (2015) can be used as a starting point for an Islamic ontology.

3 Methodology

3.1 Data Collection

This section presents an overview of gathering and extracting Qur'an and Hadith data from four datasets: LK-Hadith-Corpus[1], HT_Topics, QH_Dataset[2], and QuranOntology (Hakkoum and Raghay, 2015; Hakkoum and Raghay, 2016). These datasets include different data types, such as well-structured formats, semi-structured, and data lacking organisation known as unstructured.

3.1.1 LK-Hadith-Corpus

The Leeds University and King Saud University (LK) Hadith corpus is a well-structured bilingual Arabic-English Islamic Hadith. It was generated automatically using a Hadith segmentation tool to link each Hadith into its two main components, Isnad and Matn. This corpus contains 39,038 annotated Ahadith elicited from the six canonical Hadith books, including names and numbers of each book, chapter, and section. The LK-Hadith corpus also has over 10 million tokens. The total number of Hadith chapters is 238, excluding duplication (Altammami et al., 2020).

3.1.2 HT_Topics Dataset

The Hadith_Teaching_Topics dataset comprises more than 33 thousand Arabic-Matn documents and their related topics. The Arabic-Matn documents were extracted from LK-Hadith-Corpus (Altammami et al., 2020). The incomplete and empty

---

[1] `https://github.com/ShathaTm/LK-Hadith-Corpus`

[2] `https://github.com/ShathaTm/Quran_Hadith_Datasets/blob/main/QH_Dataset.csv`

Hadith-teaching records were eliminated, resulting in a final dataset containing 33,169 Arabic-Matn documents. The Arabic-Matn documents have been analysed using a comparison of the BERTopic technique with different Arabic transformer-based language models and topic representations to detect hidden topics from each document. ArabicBERT model achieved the best results and generated 220 topics. Consequently, each Hadith-teaching was semantically classified into a specific related topic.

3.1.3 QH_Dataset

Qur'an_Hadith_Dataset is a collection of 310 semantic relatedness pairs of Arabic Qur'an-verse and Hadith-teaching, classified as related (1) and non_related (0) pairs. The source used to build this dataset is a Fatwas website for AbdulAziz ibn Baz (a reputable Islamic scholar). The authors collected Fataws, including a Qur'an-verse and Hadith-teaching, to extract related or non-related Qur'an-verse and Hadith-teaching pairs. The limitation of this dataset is its absence of crucial data, such as the chapter names and numbers for both Qur'an-verse and Hadith-teaching. (Altammami and Atwell, 2022).

3.1.4 QuranOntology

A structured Qur'anic ontology was developed to represent and link Qur'anic knowledge from various datasets in both Arabic and English languages (Hakkoum and Raghay, 2015). It includes Qur'anic metadata and text from the Tanzil website, Qur'an descriptions from Tafsir AlJalalayn and Al-Muyasser books, Qur'anic concepts and topics discussed in Tafsir Ibn Kathir. It integrated the Semantic Qur'anic from QurSim (Sharaf and Atwell, 2012b) and the most significant Qur'anic annotations in the Qur'ana dataset (Sharaf and Atwell, 2012a).

3.2 Data Extraction and Pre-Processing

Data extraction is considered an essential step in word embeddings because of its importance in retrieving relevant information from various sources and other data repositories. To integrate relevant text data for Qur'an-verse and Hadith-teaching in QH_Dataset, we applied the following steps:

- The initial step is to extract Qur'an topics from QuranOntology corresponding to each Qur'an verse and related information such as verse numbers and chapters.

- Then, Hadith topics are meticulously extracted for each corresponding Hadith teaching from the HT_Topics dataset, and the associated details from LK-Hadith-Courpus.

- Finally, the Qur'an-Hadith-Topics (QHT) dataset is created and comprehensively combined 310 Arabic Qur'an-verse and Hadith-teaching pairs with their chapter numbers and names, Qur'an-Hadith topics, and labelled as 155 related pairs (1), and 155 non-related (0).

At the same time, preparing data is crucial for many NLP applications to enhance data quality. This process includes cleaning the raw data by identifying, correcting, and eliminating unnecessary data, such as missing and irrelevant elements. After that, the data is segmented into tokens to ensure more efficient and accurate analysis for the intended use. The topics of Qur'an-verse and Hadith-teaching pairs are then divided into 80% training and 20% testing sets.

3.3 Word Embeddings

Word embedding is an NLP technique. It represents words as dense vectors to capture semantic and syntactic relationships between texts. We employed BERT (Bidirectional Encoder Representations from Transformers) as a state-of-the-art word embedding model. BERT leverages pre-trained transformer-based language models by capturing contextual word embeddings to enhance language understanding in various NLP tasks, specifically semantic similarity.

This study uses BERT with multiple pre-trained transformer-based models to generate numerical vector representations of Qur'an-Hadith topics. The Arabic pre-trained models include: AraBERTv2 (Antoun et al., 2020), ArabicBERT (Safaya et al., 2020), CAMeL-BERT (Inoue et al., 2021), CL-AraBERT (Malhas and Elsayed, 2022), and ARBERT and MARBERT (Abdul-Mageed et al., 2021). We configured the batch size to 8, the number of epochs to 5, and the learning rate to 1e-4 for all the proposed BERT models, as a result, these settings produced the best performance in terms of accuracy and F1 score metrics.

3.4 Semantic Similarity

The semantic similarity concept has a vital significance in NLP and computational linguistics. It can support machines' understanding and process-

Models	F1 Score	Accuracy
AraBERTv2 (Antoun et al., 2020)	0.8571	0.7556
ArabicBERT (Safaya et al., 2020)	0.8981	0.8220
CAMeL-BERT (Inoue et al., 2021)	**0.9024**	0.8222
CL-AraBERT (Malhas and Elsayed, 2022)	**0.9136**	**0.8444**
ARBERT (Abdul-Mageed et al., 2021)	0.8815	0.8133
MARBERT (Abdul-Mageed et al., 2021)	**0.9085**	0.8371

Table 1: The results of the Arabic pre-trained language models.

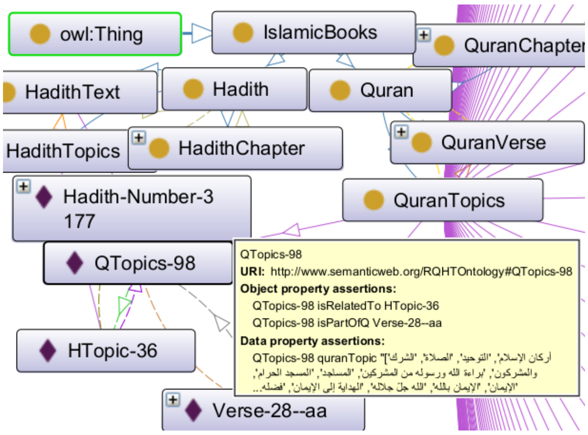
ing of human language by computing the semantic relatedness between pairs of word embedding (Rahutomo et al., 2012). Cosine similarity, one of the standard semantic relatedness metrics, is calculated to quantify similarity and ranges from -1 (entirely dissimilar) to 1 (similar).

Consequently, a cosine similarity matrix is generated for the pairs of Qur'an-Hadith-Topics numerical representations to capture the nuanced semantic relatedness between them. Then, the results of the cosine similarity are compared with the QH_Dataset labels to assess the performance of the best model semantic similarity.

3.5 Ontology Creation with Cellfie Plugin

The Cellfie Plugin[3] is a powerful Protégé tool developed to automatically enhance knowledge management and ontology engineering. It has the ability to convert structured data in spreadsheets (in formats like CSV or Excel) into web ontology language (OWL) format. It maps the spreadsheet data into ontology entities such as classes, individuals, and properties by using a rule-based mapping approach. Therefore, the outcomes of the related topics from the semantic similarity process can be linked with the Qur'an and Hadith data through an ontology, namely Related Qur'an and Hadith Topics (RQHT).

To build the RQHT ontology, we initially built the conceptual model for the QHT dataset and defined the ontology classes, data and object properties. The Cellfie plugin was then used to import the QHT spreadsheet and set mapping rules to ensure that each column is mapped to the appropriate classes or properties, and each row is accurately mapped to the corresponding individuals in the RQHT ontology. Figure 1 shows an example of the topics related to a Qur'an-verse and Hadith-teaching pair and integrated with Qur'an and Hadith knowledge.



QTopics-98
URI: http://www.semanticweb.org/RQHTOntology#QTopics-98
Object property assertions:
QTopics-98 isRelatedTo HTopic-36
QTopics-98 isPartOfQ Verse-28--aa
Data property assertions:
QTopics-98 quranTopic "['أركان الإسلام', 'التوحيد', 'الصلاة', 'الشرك
والمشركون', 'براءة الله ورسوله من المشركين', 'المساجد', 'المسجد الحرام',
'الإيمان', 'الإيمان بالله', 'الله جلّ جلاله', 'الهداية إلى الإيمان', 'فضله...

Figure 1: An example of RQHT ontology visualisation.

4 Evaluation

This section describes several methods to evaluate the semantic similarity of the word embeddings' performance on the test set, including F1 score and accuracy metrics, classifying pairs as similar or dissimilar with a threshold of 0.5. The threshold can classify topic pairs as similar if the similarity score is greater than 0.5 and dissimilar if the score is lower. These metrics comprehensively understand how effectively our method enhances and classifies relevant connections.

The F1 score provides a harmonic of precision (the accuracy of positive predictions) and recall (the true positive predictions), ranging from 0 for the worst performance to 1 for the best. Its formula is defined as follows:

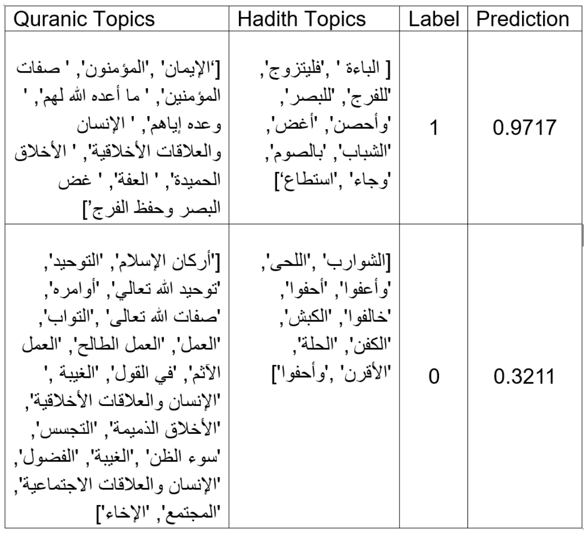
$$F1\ score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

While the accuracy measures the percentage of correctly classified instances (true positives and true negatives) out of the total number of instances, it can be calculated using the following formula:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

---

[3] `https://github.com/protegeproject/cellfie-plugin`



Quranic Topics	Hadith Topics	Label	Prediction
['الإيمان' ,'المؤمنون', ' صفات المؤمنين', ' ما أعده الله لهم', ' وعده إياهم', ' الإنسان والعلاقات الأخلاقية', ' الأخلاق الحميدة', ' العفة', ' غض البصر وحفظ الفرج']	[الباءة ' ,'فليتزوج', 'للفرج', 'للبصر', 'وأحصن', 'أغض', 'الشباب', 'بالصوم', 'وجاء' ,'استطاع']	1	0.9717
['أركان الإسلام', 'التوحيد', 'توحيد الله تعالي', 'أوامره', 'صفات الله تعالى' ,'التواب', 'العمل', 'العمل الطالح', 'العمل الآثم', 'في القول', 'الغيبة ', 'الإنسان والعلاقات الأخلاقية', 'الأخلاق الذميمة', 'التجسس', 'سوء الظن' ,'الغيبة', 'الفضول', 'الإنسان والعلاقات الاجتماعية', 'المجتمع', 'الإخاء']	['الشوارب' ,'اللحى', 'وأعفوا', 'أحفوا', 'خالفوا', 'الكبش', 'الكفن', 'الحلة', 'الأقرن' ,'وأحفوا']	0	0.3211

Figure 2: Examples of comparing the semantic similarity scores between QH_Dataset labels and CL-AraBERT model embeddings results.

Metrics	
Axioms	2497
Class count	9
Object property count	4
Data property count	6
Individual count	463

Table 2: The RQHT ontology metrics.

Regarding the RQHT ontology, it is evaluated based on various criteria: correctness, completeness, and consistency. HermiT tool is the first freely available OWL Reasoner based on a novel "hypertableau" calculus (Glimm et al., 2014).
It is employed in this work to assess the consistency of the ontology and detect subsumption links between classes.

5 Results

This section provides an analysis of the performance of the proposed methodology for linking Qur'an and Hadith topics semantically. Table 1 shows the performance of the mentioned pre-trained Arabic language models. CAMeL-BERT, CL-AraBERT, and MARBERT models performed quite similar findings in F1 score, but CL-AraBERT is the best among them with 0.9136 F1 score and 0.8444 accuracy score. The CL-AraBERT has successfully categorised the binary classification of Qur'an and Hadith topic pairs.

Figure 2 illustrates examples of the results obtained from the semantic similarity scores based on CL-AraBERT model embeddings. The predicted scores compared to the labels of the QH_Dataset. As previously mentioned, the Qur'an and Hadith topics with semantic similarity scores exceeding the threshold of 0.5 are considered related. Subsequently, the results show no significant difference and explain the effectiveness of the model embedding in capturing the semantic relatedness of the Qur'an and Hadith topics.

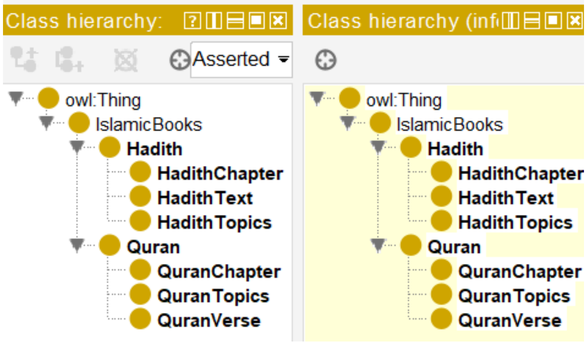


Figure 3: An example of the inferred class hierarchy using the HermiT reasoner.

Concerning RQHT ontology, Table 2 presents some of the ontology information. Also, Figure 1 illustrates the visualization of our ontology. It explains the related Qur'an and Hadith topics with their related knowledge, such as verse and Hadith numbers. Our experiment provides a consistent mapping between the classes, their instances and related topics by using HermiT reasoner. For example, the HermiT reasoner detected the inferred class hierarchy consistently, as shown in Figure 3.

6 Conclusion

In this paper, we described the experimental phases of linking Qur'an and Hadith topics and their related knowledge from various datasets. To the best of our knowledge, there is no work on linking Qur'an and Hadith topics and integrating them to related knowledge from many Qur'an and Hadith resources using word embeddings and the Cellfie Plugin tool.

This research consists of extracting the semantic similarity scores of Qur'an and Hadith topics and then comparing them to the labels of Qur'an-verse and Hadith-teaching pairs. Then, the Cellfie tool was employed to build the RQHT ontology by converting structured data in spreadsheets into OWL formats. The results of the conducted experiment have shown that the semantic similarity scores using the CL-AraBERT model are similar to the labels and obtained an F1 score of 0.91. Meanwhile, the ontology correctly links the Qur'an and Hadith topics with their related data.

For future work, we plan to extend RQHT ontology by linking more Qur'an and Hadith data from different types of datasets, such as covering more languages and question-answering datasets. This work could provide a unification of Islamic resources and could contribute to many aspects of Islamic or religious education disciplines.

Limitations

The potential limitation of this work is the small amount of QH_Dataset, related Qur'an-verse and Hadith-teaching pairs. Thus, the experiment was conducted on topics that are related to the pairs of Qur'an-verse and Hadith-teaching.

Ethics Statement

In this study, we used datasets derived from publicly accessible data resources. Including Qur'an Ontology, QH_Dataset, and LK-Hadith-Corpus. The HT_Topics dataset is built by us and will be available in the future.

Acknowledgements

The first author would like to express her deepest gratitude for her supervisor's comments. Also, she would like to thank the University of Hafr Al Batin for sponsoring her PhD studies at the University of Leeds through the Saudi Arabia Cultural Bureau (SACB).

References

Ahmed Abdelghany, Hammam Abdelaal, Abdulrahman Kamr, and Passent Elkafrawy. 2020. Doc2vec: An approach to identify hadith similarities. *Australian Journal of Basic and Applied Sciences*, 14:46–53.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Hanan A. Al-Sanasleh and Bassam H. Hammo. 2017. Building domain ontology: Experiences in developing the prophetic ontology form quran and hadith. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 223–228.

Abdullah Alsaleh, Eric Atwell, and Abdulrahman Altahhan. 2021. Quranic verses semantic relatedness using AraBERT. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 185–190, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Rusul Yousif Alsalhee and Abdulhussein Mohsin Abdullah. 2022. Building quranic stories ontology using mappingmaster domain-specific language. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(1):684–693.

Ibtisam Khalaf Alshammari, Eric Atwell, and Mohammad Ammar Alsalka. 2022. Automatic mapping of quranic ontologies using rml and cellfie plugin. In *Natural Language Processing and Information Systems*, pages 307–314, Cham. Springer International Publishing.

Menwa Alshammeri, Eric Atwell, and Mhd Alsalka. 2022. A siamese transformer-based architecture for detecting semantic similarity in the quran. *International Journal on Islamic Applications in Computer Science And Technology*, 9(4).

Menwa Alshammeri, Eric Atwell, and Mohammad Ammar Alsalka. 2021. Classifying verses of the quran using doc2vec. In *ACL Anthology*. ACL.

Shatha Altammami and Eric Atwell. 2022. Challenging the transformer-based models with a classical Arabic dataset: Quran and hadith. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1462–1471, Marseille, France. European Language Resources Association.

Shatha Altammami, Eric Atwell, and Ammar Alsalka. 2020. The arabic–english parallel corpus of authentic hadith. *International Journal on Islamic Applications in Computer Science And Technology*, 8(2).

Shatha Altammami, Eric Atwell, and Ammar Alsalka. 2021. Towards a joint ontology of quran and hadith. *International Journal on Islamic Applications in Computer Science And Technology*, 9(2):01–12.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Bendjamaa Fairouz, Taleb Nora, and Arari Amina Nouha. 2020. An ontological model of hadith texts. *International Journal of Advanced Computer Science and Applications*, 11(4).

Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. 1997. Methontology: From ontological art towards ontological engineering. In *AAAI Conference on Artificial Intelligence*.

Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. 2014. Hermit: An owl 2 reasoner. *Journal of Automated Reasoning*, 53:245–269.

Thomas R. Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5):907–928.

Aimad Hakkoum and Said Raghay. 2015. Advanced search in the qur'an using semantic modeling. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–4.

Aimad Hakkoum and Said Raghay. 2016. Semantic qa system on the qur'an. *Arabian Journal for Science and Engineering*, 41:5205–5214.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *ArXiv*, abs/2103.06678.

Rana Malhas and Tamer Elsayed. 2022. Arabic machine reading comprehension on the holy qur'an using clarabert. *Inf. Process. Manag.*, 59:103068.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. volume 4.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Abdul-Baquee Sharaf and Eric Atwell. 2012a. QurAna: Corpus of the Quran annotated with pronominal anaphora. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 130–137, Istanbul, Turkey. European Language Resources Association (ELRA).

Abdul-Baquee Sharaf and Eric Atwell. 2012b. QurSim: A corpus for evaluation of relatedness in short texts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2295–2302, Istanbul, Turkey. European Language Resources Association (ELRA).

Medical Information Extraction with Large Language Models

Raffaello Fornasiere, **Nicolò Brunello**, **Vincenzo Scotti** and **Mark James Carman**
DEIB, Politecnico di Milano
Via Ponzio 34/5, 20133, Milano (MI), Italy
raffaello.fornasiere@mail.polimi.it nicolo.brunello@polimi.it
vincenzo.scotti@polimi.it mark.carman@polimi.it

Abstract

The increase in clinical text data following the adoption of electronic health records offers benefits for medical practice and introduces challenges in automatic data extraction. Since manual extraction is often inefficient and error-prone, with this work, we explore the use of open, small-scale, Large Language Models (LLMs) to automate and improve the extraction of medication and timeline data. With our experiments, we aim to assess the effectiveness of different prompting strategies –zero-shot, few-shots, and sequential prompting– on LLMs to generate a mixture of structured and unstructured information starting from a reference document. The results show that even a zero-shot learning approach can be sufficient to extract medication information with high precision. The main issues in generating the required information seem to be completeness and redundancy. However, prompt tuning alone seems to be sufficient to achieve good results using these LLMs, even in specific domains like the medical one. Besides medical information extraction, in this work, we address the problem of explainability, introducing a line-number referencing method to enhance transparency and trust in the generated results. Finally, to underscore the viability of applying these LLM-based solutions to medical information extraction, we deployed the developed pipelines within a demo application.

1 Introduction

The rapid integration of digital technologies into healthcare systems has transformed the landscape of patient care and management. *Electronic Health Record* (EHR) systems have become pivotal in modern healthcare environments. However, as a downside, primary care physicians, for example, face a significant burden of documentation. Research indicates that family medicine physicians allocate nearly as much time to interacting with EHR systems as they do to direct patient care (Arndt et al., 2017), leading to reduced clinical efficiency and increased risk of clinician burnout.

To address these issues and automate (or semi-automate) the analysis of these documents and, thus, reduce clinicians burden, we explore the application of *Large Language Models* (LLMs) (Brown et al., 2020; OpenAI, 2023; Anil et al., 2023) as a means to enhance the functionality and efficiency of EHR systems. LLMs are the pivot of the current advancements in *Artificial Intelligence* (AI), present promising solutions for automating routine documentation, extracting information from unstructured data and supporting clinical decision-making through real-time insights from extensive medical databases.

Specifically, with this paper, we explore the application of small-scale openly-available LLMs (Touvron et al., 2023; Jiang et al., 2023; Mesnard et al., 2024) to automate the extraction of medication information and timeline data from clinical text. We evaluate LLMs performance in *zero-shot learning*, *few-shot learning* and *sequential prompting* scenarios. We selected the latter approach to guide the LLM through the multiple steps of information extractions in the cases where the information is not immediately accessible from the raw text. The objective of the evaluations is to assess the accuracy and completeness of the information LLMs extract, such as dosage, frequency, and mode of administration of a drug, as well as LLMs ability to construct patient timelines from clinical narratives. Through this work, we seek not only to deepen our understanding of the capabilities and reliability of LLM-based systems in medical contexts, but also to offer viable strategies for alleviating the documentation burden that detracts from patient-focused healthcare that can serve as possible baselines.

We divide this paper into the following sections. In Section 2, we recap the main results in information extraction. In Section 3, we describe the pipelines we developed for information extraction. In Section 4, we describe the data sets we used to evaluate our pipelines. In Section 5, we outline the experiments we conducted. In Section 6, we report and comment on the experimental results. Finally, in Section 7, we summarise our work and present possible future directions.

2 Related Works

Information Extraction (IE) is one of the main applications of *Natural Language Processing* (NLP), even outside the medical domain. Traditionally, information extraction encompasses problems like *Named Entities Recognition* (NER), *Relation Extraction* (RE) or *Aspect Classification* (AC) (Jurafsky and Martin, 2024, Chapter 19). NER involves the extraction of named entities like persons and locations, as well as time expressions and even drugs. RE is the task of classifying relations among entities, like the dosage of a specific drug. AC is the classification of events according to their internal temporal structure or temporal contour, for example, identifying whether a patient has been taking a drug before or after hospitalisation.

Initially, these problems have been approached with either rule-based systems or classification models combined with *Conditional Random Fields* (CRF) (Jurafsky and Martin, 2024, Chapter 19). Rule-based techniques are known for their precision in entity recognition or relation extraction, particularly when they are meticulously crafted to align with specific data types. These methods typically analyse sentence structures and leverage *Part-of-Speech* (PoS) tags to enhance NER. Both rule-based systems and classification models relied on hand-crafted features and lexical resources to identify medical entities (Landolsi et al., 2024). While they are less flexible and harder to scale, they perform reasonably on well-defined problems.

The advances introduced by *word embeddings* combined with sequence processing techniques based on deep learning like *Recurrent Neural Networks* (Elman, 1990; Hochreiter and Schmidhuber, 1997) and *Transformer Networks* (Vaswani et al., 2017) helped push forward significantly state of the art for IE. In fact, even now, many approaches often favour a combination of *Bi-directional Long Short-Term Memory* (BiLSTM) (an RNN variant) and CRF models or more recent *fine-tuned* Transformers (Symeonidou et al., 2019; Yang et al., 2020; Kafikang and Hendawi, 2023). Bi-directional models (both recurrent and Transformer) excel in capturing high-quality features due to their ability to account for contextual dependencies in both forward and backward directions. Meanwhile, CRF enhances the process by optimising sequence tagging with these features (Çelkmasat et al., 2022). These models exploit a *Begin-Inside-Outside* (BIO) tagging system which allows segmenting an input document into multiple pieces (delineating entity boundaries, for example) while labelling those same pieces (thus, recognising the type of entity, for example). Contextual models like RNN and Transformers play a crucial role in medical information extraction especially when pre-trained on medical texts so that they can incorporate domain knowledge (Lee et al., 2020; Landolsi et al., 2024).

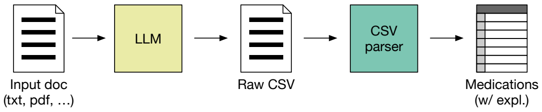
As with many other NLP tasks, LLMs have revolutionised IE as well, offering near state-of-the-art performances out of the box. The in-context learning capabilities of LLMs like *GPT-4* (OpenAI, 2023) or *Gemini* (Anil et al., 2023) have shown promising directions for biomedical NER and RE, especially in scenarios lacking labelled data. Despite these advancements, these LLMs still do not outperform consistently smaller models fine-tuned on task-specific datasets yet (Tian et al., 2023). Additionally, the use of LLMs in IE faces several challenges: For instance, the generative nature of these models may alter the phrasing of recognised entities or predicted relationships, complicating the verification process. Moreover, these LLMs are prone to hallucinations that may lead to the generation of entities and relationships that appear plausible but are not factually accurate. Furthermore, finding suitable prompts for NER and RE tasks can be difficult. These issues underscore the need for further research to explore and develop more effective methods for effectively using LLMs in IE. In this paper, we focus on the medical domain, and we explore solutions for medical IE from patients' records.

3 Methodology

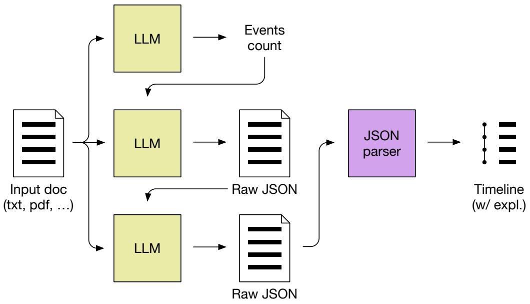
In this section, we describe the two pipelines we propose for medical information extraction from clinical documents. We provide an overview of the pipelines and information extraction approaches in Section 3.1, and then we provide additional details on the explainability in Section 3.2. Finally, we provide practical details on the two tasks of *medication extraction* and *timeline extraction* in

Section 3.3.

3.1 Information extraction overview



(a) Medication.



(b) Timeline.

Figure 1: Information extraction pipelines.

The pipeline (depicted in Figure 1) we propose is designed to give a reference medical document (e.g., a discharge letter) as input to an LLM and use the LLM to extract the desired information from the document according to user-provided instructions. As premised, in this work, we focus on medication extraction (i.e., the extraction of information about the drug regime for a patient) and timeline extraction (i.e., the sequence of events characterising a patient's clinical history). The pipeline includes the generation of the required information from the text, the parsing of the generated unstructured text and the rendering of the structured data selected from parsing. An additional passage that enriches our pipeline is that of explainability to justify the generated content.

We considered three different approaches to deal with the information extraction, independently of the actual task:

Zero-shot learning where we provide the LLM with the instructions of the task to complete and a description of the expected output;

Few-shots learning where we provide the same instruction as zero-shot learning, but before asking for the current sample, we append some examples of input and expected output to help guide the generation process;

Sequential prompting where we have the same settings of zero-shot learning, but we break down the task into multiple steps to help the LLM build the solution one piece at a time and keep it aligned with the desired behaviour.

All the aforementioned pieces are deployed as part of a web app demonstrating AI solutions for healthcare (see Figure 2). The demo is agnostic of the underlying LLM. It allows the loading of a reference document and separately generating the table with the medication information, generating the timeline and asking questions to the chatbot about the document. The raw outputs of information extraction are parsed to be converted into structured information and then displayed on the demo web page.

3.2 Explainability

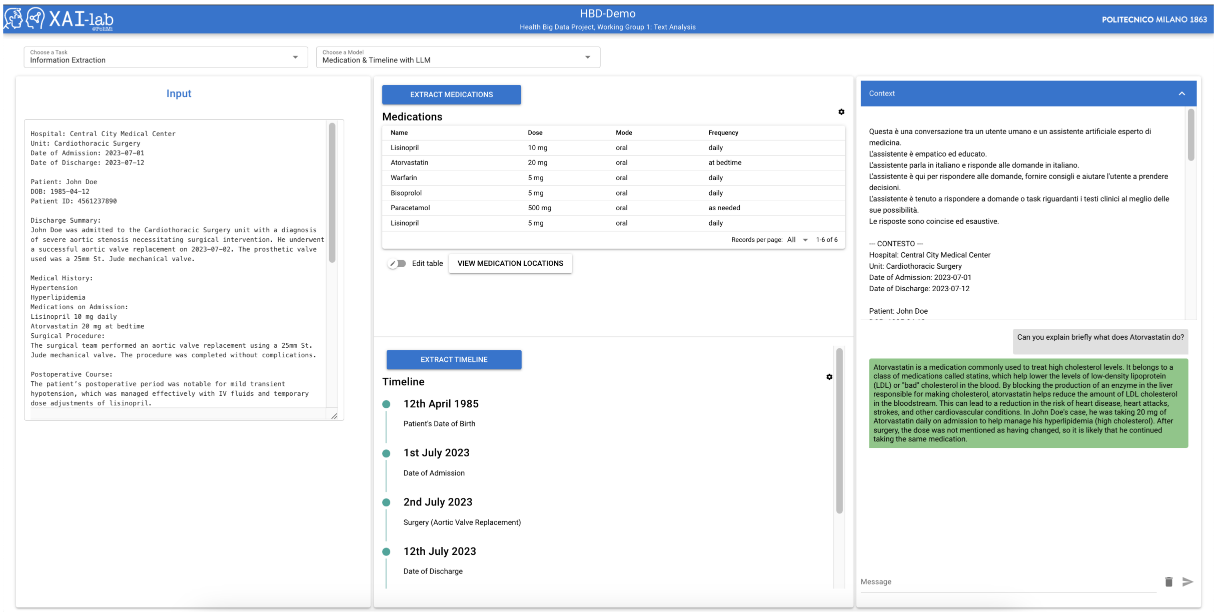
Explainability has become a more and more important step in developing and deploying deep learning-based systems. Explainability helps in understanding where model predictions come from. When it comes to healthcare, the attention to this information is even more crucial.

In our tool, we suggested a simple yet effective solution to explain the results of information extraction. We have a separate pipeline ingesting the clinical document to analyse decorated with the rows numbers. In this way, we can interrogate the LLM automatically asking to point out the number of the row connected to a specific extraction (e.g., where is a specific drug mentioned or where is a specific event mentioned).

This additional explanation can be useful for the clinician. In fact, on one side, having an explanation helps ground the predicted information. On the other side, it helps spot possible errors due to LLM faults, preventing the misinformation of the clinician.

3.3 Medications and Timeline extraction

As premised, in the deployed demo, we approach both medication information extraction and timeline extraction. We approach medication extraction with a zero-shot learning approach and timeline extraction with sequential prompting. We selected these approaches given the results of the experiments we conducted. In both tasks we use the model in *assistant chatbot* format (Scotti et al., 2024), composing a sequence of messages to explain and solve the tasks.



```
Hospital: Central City Medical Center
Unit: Cardiothoracic Surgery
Date of Admission: 2023-07-01
Date of Discharge: 2023-07-12

Patient: John Doe
DOB: 1985-04-12
Patient ID: 4561237890

Discharge Summary:
John Doe was admitted to the Cardiothoracic Surgery unit with a diagnosis
of severe aortic stenosis necessitating surgical intervention. He underwent
a successful aortic valve replacement on 2023-07-02. The prosthetic valve
used was a 25mm St. Jude mechanical valve.

Medical History:
Hypertension
Hyperlipidemia
Medications on Admission:
Lisinopril 10 mg daily
Atorvastatin 20 mg at bedtime
Surgical Procedure:
The surgical team performed an aortic valve replacement using a 25mm St.
Jude mechanical valve. The procedure was completed without complications.

Postoperative Course:
The patient's postoperative period was notable for mild transient
hypotension, which was managed effectively with IV fluids and temporary
dose adjustments of lisinopril.
```

Medications

Name	Dose	Mode	Frequency
Lisinopril	10 mg	oral	daily
Atorvastatin	20 mg	oral	at bedtime
Warfarin	5 mg	oral	daily
Bisoprolol	5 mg	oral	daily
Paracetamol	500 mg	oral	as needed
Lisinopril	5 mg	oral	daily

Records per page: All 1-6 of 6

Timeline

12th April 1985
Patient's Date of Birth

1st July 2023
Date of Admission

2nd July 2023
Surgery (Aortic Valve Replacement)

12th July 2023
Date of Discharge

Questa è una conversazione tra un utente umano e un assistente artificiale esperto di medicina.
L'assistente è empatico ed educato.
L'assistente parla in italiano e risponde alle domande in italiano.
L'assistente è qui per rispondere alle domande, fornire consigli e aiutare l'utente a prendere decisioni.
L'assistente è tenuto a rispondere a domande o task riguardanti i testi clinici al meglio delle sue possibilità.
Le risposte sono coincise ed esaustive.

--- CONTESTO ---
Hospital: Central City Medical Center
Unit: Cardiothoracic Surgery
Date of Admission: 2023-07-01
Date of Discharge: 2023-07-12

Patient: John Doe

Can you explain briefly what does Atorvastatin do?

Atorvastatin is a medication commonly used to treat high cholesterol levels. It belongs to a class of medications called statins, which help lower the levels of low-density lipoprotein (LDL) or "bad" cholesterol in the blood. By blocking the production of an enzyme in the liver responsible for making cholesterol, atorvastatin helps reduce the amount of LDL cholesterol in the bloodstream. This can lead to a reduction in the risk of heart disease, heart attacks, strokes, and other cardiovascular conditions. In John Doe's case, he was taking 20 mg of Atorvastatin daily on admission to help manage his hyperlipidemia (high cholesterol). After surgery, the dose was not mentioned as having changed, so it is likely that he continued taking the same medication.

Figure 2: Demo tool using the pipeline to extract medications and medical events from a reference document.

Concerning medication extraction, we task the model to extract all the information at once from a reference document. We provide the reference document to the model as part of the system message, we then append a user message describing the task, and, finally, we force the answer of the assistant to start with the raw text content of the CSV file we want as output. We get the LLM to generate starting from these messages. The CSV table contains the following information about the medication: *name*, *dose* (the specified amount of medication), *mode* (intended as mode of administration), *frequency* (how many times or how often to take the medication), *line* (line in the text where the medication information is mentioned, for explainability).

Concerning the timeline extraction, we follow a sequential prompting approach. We broke down the task into three steps: counting the events mentioned in the document, generating a JSON array with the chronologically ordered events and generating the line number for each event on the array. As for the other task, we provide the reference document as part of the system message and then we alternate user messages with the instructions for the current step and model responses for that step. The elements of the JSON array with the chronological order are *dateValue* (date in the format `"YYYY-MM-DD"`), *dateString* (the string mentioning the date as it appeared in the original document) and *event* (a short description of what the event that occurred at that time point). Each element is decorated in the last step with the line number for explainability. We found empirically these steps to be the most effective to generate the timeline.

4 Data

One of the challenges of working in the healthcare domain is gathering usable data. Given the nature of the task, we focused on finding data sets containing similar samples to what the model would encounter in real-world scenarios. For this project, we resorted to two existing data sets, one for medication extraction and one for timeline extraction (we describe them respectively in Section 4.1 and Section 4.1), and we generated a third additional data set synthetically (we describe this third data set in Section 4.1).

4.1 N2C2

The *National NLP Clinical Challenges* (N2C2) data set (Uzuner et al., 2010) is a collection of 1243 de-identified discharge summaries from *Partners Healthcare*. This data set was released as part of a medical annotation challenge. In the challenge, participants were tasked with extracting medication information from these summaries and collectively provided annotations for 251 documents. The dataset focuses on the identification of medications and medication-related information, including *dosages*, *routes* (i.e., models of administration), *frequencies*, *durations*, and *reasons* for administration.

Listing 1: Example of the N2C2 input document with numbered lines.

```
41 HASSEL , EDWARDO D. , M.D
42 on order for NEPHROCAPS PO ( ref 12327843 )
43 POTENTIALLY SERIOUS INTERACTION: SIMVASTATIN NIACIN ,
44 VIT. B-3 Reason for override: home regimen
45 Previous override information:
46 Override added on 4/29/04 by GALIPEAU , ENRIQUE R. , M.D.
47 DEFINITE ALLERGY ( OR SENSITIVITY ) to HMG CoA REDUCTASE
48 INHIBITORS Reason for override : md aware , home regimen
49 IMDUR ( ISOSORBIDE MONONIT.( SR ) ) 30 MG PO QD
50 Food/Drug Interaction Instruction
51 Give on an empty stomach ( give 1hr before or 2hr after
```

Listing 2: Example of the N2C2 output labels.

```
m="nephrocaps" 42:3 42:3||do="nm"||mo="po" 42:4 42:4||f="nm"||du="nm"||r="nm"||ln="list"
m="niacin" 43:5 43:5||do="nm"||mo="nm"||f="nm"||du="nm"||r="nm"||ln="list"
m="simvastatin" 43:3 43:3||do="nm"||mo="nm"||f="nm"||du="nm"||r="nm"||ln="list"
m="vit.\ b-3" 44:0 44:1||do="nm"||mo="nm"||f="nm"||du="nm"||r="nm"||ln="list"
m="imdur ( isosorbide mononit.( sr ) )" 49:0 49:6||do="30 mg" 49:7 49:8||mo="po" 49:9 49:9||f="qd" 49:10 49:10||du="nm"||r="nm"||ln="list"
```

The annotations provide the precise location of each piece of information within the discharge summaries, facilitating the development and evaluation of NLP systems for medication information extraction. We report examples of input document (chunk) and corresponding annotations respectively in Listing 1 and Listing 2. As can be evicted by the annotations, the target data contain all the desired details and their reference within the document.

I2B2

The *Informatics for Integrating Biology and the Bedside* (I2B2) data set (Sun et al., 2013), released as part of the homonymous project, consists of 310 discharge summaries annotated with temporal information. This data set was created to facilitate the development and evaluation of NLP systems for temporal reasoning in clinical text.

Listing 3: Example of the N2C2 input document with numbered lines.

```
41 Admission Date :
42 09/29/1993
43 Discharge Date :
44 10/04/1993
45 HISTORY OF PRESENT ILLNESS :
46 The patient is a 28-year-old woman who is HIV positive for two years .
47 She presented with left upper quadrant pain as well as nausea and vomiting which is a long-standing complaint .
48 She was diagnosed in 1991 during the birth of her child .
49 She claims she does not know why she is HIV positive .
```

Listing 4: Example of the I2B2 output labels in XML format.

```
<timex3 id="T0" start="18" end="28" text="09/29/1993" type="DATE" val="1993-09-29" mod="NA" />
<timex3 id="T13" start="2249" end="2271" text="the day of discharge ." type="DATE" val="1993-10-04" mod="NA" />
<timex3 id="T3" start="290" end="294" text="1991" type="DATE" val="1991" mod="NA" />
```

The annotations focus on three key aspects. *Events*, which include clinical concepts (problems, tests, treatments), clinical departments, evidential information (source of information), and occurrences (e.g., admissions and transfers). Each event is further categorised by type, polarity (positive or negated), and modality (factual, proposed, conditional, or possible). *Temporal expressions*, which include dates, times, durations, and frequencies, normalised to the ISO8601 standard. Each temporal expression is characterised by its type, value, and modifier (exact or approximate). *Temporal Relations* (TLinks), which describe the relationships between events and temporal expressions, indicating whether one occurred before, after, or overlapped with another. We report examples of input document (chunk) and corresponding annotations respectively in Listing 3 and Listing 4. As for the previous data set, the XML entries containing the labels are annotated also with the position of the information withing the source document.

Synthetic Data Set

Given the reduced size of the I2B2 data set, we resorted to ChatGPT to generate some additional data. We refer to this as *synthetic data set* (SD). We prompted ChatGPT 4 to create discharge summaries in both English and Italian[1], along with corresponding annotations for relevant medical information. We report examples of input document (chunk) and corresponding annotations in Listing 5. Differently from the previous two data sets, we have a single entry containing both the input document and the target labels, without explicit annotations of the position of the information within the document (it would have been unrelialable to use ChatGPT annotations for this information, which we can extract searching the matching substrings in the source document)

In general synthesising data offers several advantages, like *personalisation*, *privacy* and *control*. From the personalisation perspective, we have that the content and style of the generated summaries can be tailored to specific requirements, allowing for the creation of diverse and representative samples. Concerning privacy, since the data is synthetic, it inherently avoids privacy concerns associated with real patient data. Finally, about control, we have that the generation process allows for precise control over the types of medical information included, enabling targeted testing of specific extraction challenges. However, it's important to acknowledge that synthetic data may not fully capture the nuances and complexities of real-world clinical documentation. While it serves as a valuable resource for preliminary testing and development, its limitations should be considered when interpreting results and generalising findings to real-world scenarios.

Listing 5: Example of the SD input dcoments and output annotation in JSON format.

```
{
  "text": "**Discharge Summary
      :**\n\nPatient: Mark Johnson
        \nAge: 38  \nAdmission
      Date: 03-20-2024  \
      nDischarge Date: 03/28/24  \
      n\nPatient History:\nMr.
      Mark Johnson, a 38-year-old
      male, was admitted to our
      facility on March 20, 2024,
      presenting with complaints
      of abdominal pain, nausea,
      and jaundice. He has a past
      medical history ...",
  "annotations": [
    {
      "text": "March 20, 2024",
      "date_value": "2024-03-20"
    },
    {
      "text": "March 24, 2024",
      "date_value": "2024-03-24"
    },
    {
      "text": "eight days",
      "date_value": "2024-03-28"
    },
    ...
  ]
}
```

5 Experiments

In this section, we detail the experiments we run to evaluate our pipelines for medication extraction (Section 5.1) and timeline extraction (Section 5.2). In all the experiments we conducted, we worked with *Mistral 7B* (Jiang et al., 2023), using this LLM as the core of the information extraction system.

---

[1] We worked with Italian documents to fit the requirements of the project founding this work; for further details, refer to the acknowledgements at the end of this paper.

5.1 Medication Extraction

In the first set of experiments, we focused on medication extraction from clinical texts. The primary objective is to evaluate the models' ability to extract medication details such as dosage, mode, and frequency from unstructured medical documents, like discharge letters. For this task, we focused on the N2C2 data set. We evaluated the LLM capabilities with different approaches: zero-shot learning, few-shots learning (using 2 examples) and sequential prompting. We conducted the evaluation using the standard metrics: precision, recall, and F_1 score.

Initially, we considered two variants of this task: looking for *full medications* (i.e., we asked the LLM to generate all the medication details: name, dosage, mode, and frequency) or not (i.e., we asked the LLM to generate only the name of the medication). However, as we explain better in Section 6.1, working with full medication yields poor results, as we noticed immediately in the early experiments with a zero-shot learning approach. To measure the metrics in the full medications case, we considered a single string containing all the details, and to measure a match, we standardised the target string and the generated one by removing all spaces and special characters.

Concerning the input and output format, we considered multiple alternatives as well. We explored having as input the whole document to analyse or only a relevant chunk, this approach is helpful with particularly long documents. Moreover, we explored two different output formats: JSON and CSV; in both cases we had the LLM generate directly the raw JSON or CSV strings.

5.2 Timeline extraction

In the second set of experiments, we focused on extracting patient timelines from clinical texts in order to highlight all the relevant events. In this case, we focused only on evaluating the model's capabilities in extracting correctly formatted dates. In fact, from early explorations, we noticed that this task was already challenging as the LLM often deviated from the target format. For this task, we used the I2B2 data set and the synthetic data set. We conducted the evaluation using the standard metrics: precision, recall, and F_1 score.

As for the previous experiment, we evaluated the LLM capabilities with zero-shot learning, few-shots learning (using 4 examples) and sequential prompting approaches. Concerning the input and output format, similar to medication extraction, we considered alternative approaches. As before, we explored using the whole document as input or only a relevant chunk. For the output, we worked only in JSON format and we converted all dates in `"YYYY-MM-DD"` format.

6 Results

In this section, we present and comment on the results of the experiments on medication extraction (Section 6.1) and timeline extraction (Section 6.2). In both cases, we do not compare with the reference baselines coming with the data sets since we approach the evaluations differently and we compute different metrics.

6.1 Medication Extraction

We report the results of this first task of medication extraction in Table 1. Results on precision focusing on the medication name are satisfying, meaning that the model is missing very few medications from the documents. However, the low recall and, subsequent, low F_1 scores hint that the model is often generating information that is not part of the original document. Moreover, results using full medication information are consistently lower, indicating that, as expected, extracting detailed information is harder than simply identifying the medication.

The experiments with zero-shot approach showed that the LLM is not capable of extracting all the medication information just from the instructions. Looking at the generated output, we noticed that sticking to the target output format was difficult, and even output post-processing and string normalisation were not sufficient to match the target and predicted output. CSV format seems to be harder to get to work independently of the target being name only or full medication information.

From the results of the few-shots approach and sequential approach, there seems to be no clear solution for the output format. In fact, depending on the approach, generating CSV or JSON output seems to yield the best results. Concerning the difference between the approaches, there is not clear difference between zero-shot and sequential approaches. Few-shots approach does not improve significantly over the other approaches over precision, but improves the recall and, thus, the F_1.

Approach	Format	Chunked docs	Full medication	Precision	Recall	F$_1$
zero-shot	JSON	✗	✗	0.964	0.392	0.513
		✗	✓	0.446	0.115	0.181
	CSV	✗	✗	0.557	0.453	0.498
		✗	✓	0.418	0.217	0.278
few-shots	JSON	✗	✗	0.885	0.479	0.606
		✗	✓	0.364	0.109	0.166
		✓	✗	**0.965**	0.547	**0.683**
		✓	✓	0.616	0.243	0.342
	CSV	✗	✗	0.857	0.546	0.660
		✗	✓	0.366	0.136	0.198
		✓	✗	0.837	0.526	0.636
		✓	✓	0.380	0.160	0.224
sequential	JSON	✗	✗	0.961	0.358	0.512
		✗	✓	0.597	0.134	0.217
	CSV	✗	✗	0.808	0.318	0.442
		✗	✓	0.288	**0.550**	0.378

Table 1: Results on N2C2 for medication extraction (bold values correspond the best score).

Dataset	Approach	Chunked docs	Precision	Recall	F$_1$
I2B2	zero-shot	✗	0.811	0.589	0.651
	few-shots	✗	0.803	**0.794**	**0.790**
		✓	**0.954**	0.592	0.701
	sequential	✗	0.757	0.644	0.660
SD	zero-shot	✗	0.949	0.806	0.861
	few-shots	✗	0.926	**0.917**	0.916
		✓	**0.975**	0.898	**0.931**
	sequential	✗	0.966	0.898	0.926

Table 2: Results on I2B2 and Synthetic Data (SD) for timeline extraction (bold values correspond the best score for each data set).

6.2 Timeline extraction

We report the results on this second task of timeline extraction in Table 2. As can be seen, the results are good, yet there is a lot of space for improvement. Results on the synthetic data are always better than those on the I2B2 data set.

Comparing the results of zero-shot and few-shots learning, we can see that in most cases, using the few-shots approach helped significantly improve the results on recall and, thus, F_1. The higher results on chunked documents seem to indicate that, in this case, using longer documents negatively affects the ability to extract the time information.

Both sequential prompting and zero-shot work without reference examples, yet sequential prompting performed in terms of recall and F_1, and performed comparably to the few-shots approach. This hints that the sequential approach helped the LLM capture better the target task and output format.

7 Conclusion

In this paper, we showed how we approached the problem of medical information and events extraction using LLMs. The results of the conducted experiments highlight the potential of these LLMs for automating the extraction of this information from clinical texts. The performance of these models resulted sufficiently robust for practical application in real-world settings, though there is still room for further improvements. To complete the proposed pipelines and make them more reliable, we provided also an explanation tool.

Concerning the evaluations, the LLMs exhibited significantly better performance in few-shot learning settings when compared to zero-shot learning ones, achieving, as expected, higher precision, recall, and F1 scores. However, it is important to point out that the effectiveness of the LLM varied significantly depending on factors such as the chosen output format (JSON vs CSV). For instance, although the models are capable of adapting to the requested output format, it remains unclear which format yields the most effective results. While, in some cases, the performance we achieved is suitable for practical application, these fluctuations pinpoint a challenge taht highlights the need for better models before moving to real-world applications of the LLM technology for healthcare.

To improve the overall pipeline robustness and utility, we will be working on minimising the LLM's sensitivity to minor variations in prompts, for example, working on our own fine-tuning for chatbot assistant or instruction following rather than resorting to existing solutions. Similarly, we are interested in exploring alternative evaluation metrics that assess the semantic accuracy of the extracted information, rather than relying solely on string matching. We expect that advancements in these two directions will better gauge the practical applicability and effectiveness of LLMs in processing clinical texts. At the same time, to expand the tool capabilities, we are interested in exploring more complex scenarios, where the information to extract is scattered across multiple documents, which represent a more challenging task also from the explainability perspective.

Limitations

In this paper, we mainly focused on the development and deployment of the pipeline, rather than exhaustive experiments. The first limitation is in the choice of the LLM: as for now, we evaluated the results using only Mistral 7B. A proper evaluation would require exploring other openly accessible models of the same and different sizes and closed-access models to have a reference for the comparisons. The second limitation is the size of the available data sets, which we can consider small if compared with data sets for other information extraction tasks, thus the results may be subject to high variance.

Ethics Statement

The authors do not foresee any considerable risks associated with the work presented in this paper. In principle, the presented framework is intended for information extraction from medical documents as it is thought to be used by clinicians (or other similar experts) who have authorised access to the target documents. The authors pledge to make the source code publicly available to ensure the reproducibility of the experiments.

Acknowledgements

This work has been supported by the Health Big Data project, funded by the Italian Ministry of Economy and Finance and coordinated by the Italian Ministry of Health, and results from the interaction with numerous knowledgeable researchers, clinicians, and other field experts from the Politecnico di Milano and the IRCCSs (Istituto di ricovero e cura a carattere scientifico). The work was also partially supported by the FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence).

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Brian G. Arndt et al. 2017. Tethered to the ehr: Primary care physician workload assessment using ehr event log data and time-motion observations. *Annals of Family Medicine*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Gökberk Çelkmasat, Muhammed Enes Aktürk, Yunus Emre Ertunç, Abdul Majeed Issifu, and Murat Can Ganiz. 2022. Biomedical named entity recognition using transformers with bilstm + CRF and graph convolutional neural networks. In *International Conference on INnovations in Intelligent SysTems and Applications, INISTA 2022, Biarritz, France, August 8-12, 2022*, pages 1–6. IEEE.

Jeffrey L. Elman. 1990. Finding structure in time. *Cogn. Sci.*, 14(2):179–211.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Dan Jurafsky and James H. Martin. 2024. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (3rd edition). Draft.

Maryam Kafikang and Abdeltawab M. Hendawi. 2023. Drug-drug interaction extraction from biomedical text using relation biobert with BLSTM. *Mach. Learn. Knowl. Extr.*, 5(2):669–683.

Mohamed Yassine Landolsi, Lobna Hlaoua, and Lotfi Ben Romdhane. 2024. Extracting and structuring information from the electronic medical text: state of the art and trendy directions. *Multim. Tools Appl.*, 83(7):21229–21280.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Vincenzo Scotti, Licia Sbattella, and Roberto Tedesco. 2024. A primer on seq2seq models for generative chatbots. *ACM Comput. Surv.*, 56(3):75:1–75:58.

Weiyi Sun, Anna Rumshisky, and Özlem Uzuner. 2013. Annotating temporal information in clinical narratives. *J. Biomed. Informatics*, 46(6):S5–S12.

Anthi Symeonidou, Viachaslau Sazonau, and Paul Groth. 2019. Transfer learning for biomedical named entity recognition with biobert. In *Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems co-located with 15th International Conference on Semantic Systems (SEMANTiCS 2019), Karlsruhe, Germany, September 9th - to - 12th, 2019*, volume 2451 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C. Comeau, Rezarta Islamaj, Aadit Kapoor, Xin Gao, and Zhiyong Lu. 2023. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *CoRR*, abs/2306.10070.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Özlem Uzuner, Imre Solti, and Eithon Cadag. 2010. Extracting medication information from clinical text. *J. Am. Medical Informatics Assoc.*, 17(5):514–518.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xi Yang, Jiang Bian, William R. Hogan, and Yonghui Wu. 2020. Clinical concept extraction using transformers. *J. Am. Medical Informatics Assoc.*, 27(12):1935–1942.

7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)

October 19-20, 2024

All times are according to time (GMT)

Saturday, Oct. 19, 2024 08:00 – 16:00 (GMT)

08:00-08:15	*Opening session* **Dr. Mourad Abbas and Dr. Abed Alhakim Freihat**
08:15 – 08:35	*Keynote* **:** *An Introduction to Large Language Models* **Prof. Gérard Chollet**, CNRS, France.
08:45 – 10:45	*Oral Session 1: Large Language Models* **Chair: Prof. Gérard Chollet**, CNRS, France.
08 :45 – 09:00	**Introducing wwm-german-18k - Can LLMs Crack the Million? (Or Win at Least 500 Euros?)** Matthias Aßenmacher (LMU Munich)*; Luis Karrlein (LMU Munich); Philipp Schiele (LMU Munich); Christian Heumann (LMU Munich)
09:00 – 09:15	**Design and Comparison of Arabic Negotiation Bots Using LLMs versus Seq2Seq Models with Reinforcement Learning** Ahmad Hajj (University of Wisconsin-Madison); Yasmine A Abu Adla (American University of Beirut)*; Samah Albast (AUB); Hazem Hajj (AUB); Shady Elbassuoni (AUB); Wassim El Hajj (AUB); Khaled Shaban (Qatar University)
09:15 – 09:30	**SG-RAG: Multi-Hop Question Answering With Large Language Models Through Knowledge Graphs** Yücel Saygın (Sabancı Universitesi)*; Ahmmad O. M. Saleh (Sabanci University); Gokhan Tur (University of Illinois Urbana-Champaign)
09:30 – 09:45	**Enhancing LLM-based Arabic Negotiation by Fine Tuning on Dialogue Shortcomings** Yasmine A Abu Adla (American University of Beirut)*; Hazem Hajj (AUB); Shady Elbassuoni (AUB); Khaled Shaban (Qatar University); Wassim El Hajj (AUB)
09:45 – 10:00	**Medical Information Extraction with Large Language Models** Raffaello Fornasiere (DEIB, Politecnico di Milano); Nicolò Brunello (DEIB, Politecnico di Milano); Vincenzo Scotti (DEIB, Politecnico di Milano)*; Mark Carman (Politecnico di Milano)
10:00 – 10:15	**Personalised Abusive Language Detection Using LLMs and Retrieval-Augmented Generation** Tsungcheng Yao (Griffith University Australia)*; Ernest Foo (Griffith University Australia); Sebastian Binnewies (Griffith University)
10:15 – 10:30	**Resolving LLM-Generated Gender Biases at Transformer Inference Time with Novel K-Explorers Neural Network Traversal (KeNNT)** Hanav Modasiya (Santa Clara High School)*
10:30 – 10:45	**Investigating Gender Bias in Large Language Models Through Text Generation** Shweta Soundararajan (Technological University Dublin)*; Sarah Jane Delany (TU Dublin)

10:45 – 11:45	*Oral Session 2 : Information extraction and Summarization* **Chair: Prof. Yücel Saygın**, Sabancı Universitesi, Turkey.
10:45 – 11:00	**CliqueCorex: A Self-supervised Clique-based Anchored Topic Model** Sami Diaf (Universität Hamburg)*
11:00 – 11:15	**Large-scale Summarization of Chat Transcripts in the Absence of Annotated Summaries** Pratik K Biswas (Verizon AI&D)*
11:15 – 11:30	**Conversational Exploratory Search of Scholarly Publications Using Knowledge Graphs** Phillip Schneider (Technical University of Munich)*; Florian Matthes (Technical University of Munich)
11:30 – 11:45	**Linking Quran and Hadith Topics in an Ontology using Word Embeddings and Cellfie Plugin** Ibtisam Khalaf Alshammari (University of Leeds)*; Eric Atwell (University of Leeds); Mhd Ammar Alaska (University of Leeds)
11:45 – 12:45	**Break**
12:45 – 14:00	*Oral Session 3: Human-Machine Interaction and Conversational AI* **Chair: Prof. Hend Al-Khalifa,** King Saud University, KSA.
12:45 – 13:00	**A Hybrid Retrieval Approach for Advancing Retrieval-Augmented Generation Systems** Nguyen Nam Doan (Maastricht University)*; Aki Härmä (Maastricht University); Remzi Celebi (Maastricht University); Valeria Gottardo (NLC Health Ventures)
13:00 – 13:15	**Context-Aware Question Answering in Urdu** Samreen A Kazi (Institute of Business Administration)*; Shakeel Ahmed Khoja (Institute of Business Administration)
13:15 – 13:30	**Human and Machine: Language Processing in Translation Tasks** Hening Wang (Universität Tübingen); Leixin Zhang (University of Tübingen)*; Ondřej Bojar (Charles University)
13:30 – 13:45	**Semantically Enriched Text Generation for QA through Dense Paraphrasing** Timothy Obiso (Brandeis University)*; Bingyang Ye (Brandeis University); Kyeongmin Rim (Brandeis University); James Pustejovsky (Brandeis University)
13:45 – 14:00	**Sawaal: A Framework for Automatic Question Generation in Urdu** Maria Rahim (Institute of Business Administration)*; Shakeel Ahmed Khoja (Institute of Business Administration)
14:00– 16:00	*Oral Session 4 : Advances in Native Language Identification and Text Classification* **Chair: Dr. Daniel Braun**, University of Twente, Netherlands.
14:00 – 14:15	**Native language Identification for Arabic Language Learners using Pre-trained Language Models** Mohamed Amine CHERAGUI (Ahmed Draia University)*; Mourad Abbas (High Council of Arabic); Mohammed Mediani (United Arab Emirates University)
14:15 – 14:30	**Leveraging Annotator Disagreement for Text Classification** Jin Xu (University of Twente); Mariët Theune (University of Twente); Daniel Braun (University of Twente)*
14:30 – 14:45	**Detecting ChatGPT-Generated Text with GZIP-KNN: A No-Training, Low-Resource Approach** Matthias Berchtold (SUPSI); Sandra Mitrovic (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale)*; Davide Andreoletti (SUPSI); Daniele Puccinelli (SUPSI); Omran Ayoub (SUPSI)
14:45 – 15:00	**Dual-Task Learning for AI-Generated Medical Text Detection and Named Entity Recognition** Saja B. Al-Dabet (United Arab Emirates University)*; Ban Alomar (United Arab Emirates University); Sherzod R Turaev (United Arab Emirates University); Abdelkader Belkacem (United Arab Emirates University)

15:00 – 15:15	**Efficient Few-shot Learning for Multi-label Classification of Scientific Documents with Many Classes** Tim Schopf (Technical University of Munich)*; Alexander Blatzheim (Technical University of Munich); Nektarios Machner (Technical University of Munich); Florian Matthes (Technical University of Munich)
15:15 – 15:30	**Asking the Right Questions: Exploiting Hidden Interactions in a Generative Framework for Multilingual, Multitask Classification** Sebastian-Antonio Toma (Technical University of Cluj-Napoca)*; Camelia Lemnaru (Technical University of Cluj-Napoca); Vlad Andrei Negru (Technical University of Cluj-Napoca); Rodica Potolea (Technical University of Cluj-Napoca)
15:30 – 15:45	**Native Language Identification Improves Authorship Attribution** Ahmet Yavuz Uluslu (University of Zurich)*; Gerold Schneider (University of Zurich); Can Yildizli (PRODAFT)
15:45 – 16:00	**Analyzing Politeness in Arabic Tweets: A Preliminary Study** Hend Al-Khalifa (King Saud University)*; Maria BOUNNIT (iWAN Research Group); Nadia GHEZAIEL (University of Hail)

Sunday, Oct. 20, 2024 08:30 – 16:00 (GMT)

08:30– 10:15	*Oral Session 5: Audio, ASR, and TTS* **Chair: Prof. Nicolas Ballier**, Université Paris Cité, France.
08 :30 – 08:45	**On Barriers to Archival Audio Processing** Peter R Sullivan (UBC)*; Muhammad Abdul-Mageed (iSchool@UBC)
08:45 – 09:00	**Comparative Analysis of Modality Fusion Approaches for Audio-Visual Person Identification and Verification** Aref Farhadipour (University of Zurich)*; Masoumeh Chapariniya (University of Zurich); Teodora Vukovic (University of Zurich); Volker Dellwo (University of Zurich)
09:00 – 09:15	**Generative Adversarial Network based Neural Vocoder for Myanmar End-to-End Speech Synthesis** Aye Mya Hlaing (University of Computer Studies, Yangon)*; Win Pa Pa (University of Computer Studies, Yangon)
09:15 – 09:30	**Double Decoder: Improving latency for Streaming End-to-end ASR Models** Riqiang Wang (Dialpad)*; Shreekantha Nadig (Dialpad); Daniil Kulko (Dialpad); Simon Vandieken (Dialpad); Chia-tien Chang (Dialpad); Seyyed Saeed Sarfjoo (Dialpad); Jonas Robertson (Dialpad)
09:30 – 09:45	**Probing Whisper Predictions for French, English and Persian Transcriptions** Nicolas Ballier (Université Paris Cité)*; Léa Burin (Université Paris Cité); Behnoosh Namdarzadeh (Université Paris Cité); Sara B Ng (University of Washington); Richard Wright (University of Washington); Jean-Baptiste Yunès (Université Paris Cité)
09:45 – 10:00	**Thonburian Whisper: Robust Fine-tuned and Distilled Whisper for Thai** Zaw Htet Aung (Mahidol University)*; Thanachot Thavornmongkol (Looloo Technology); Atirut Boribalburephan (Looloo Technology); Vittavas Tangsriworakan (Mahidol University); Knot Pipatsrisawat (Looloo Technology); Titipat Achakulvisut (Mahidol University)
10:00 – 10:15	**Text-to-Speech in Azerbaijani Language via Transfer Learning in a Low Resource Environment** Dzhavidan Zeinalov (Kapital Bank)*; Bugra Sen (Kapital Bank); Firangiz Aslanova (Kapital Bank)
10:15 – 10:30	**Break**
10:30 – 12:15	*Oral Session 6 : Speech emotion recognition and speaker verification and diarization* **Chair: Dr. Mohammed Mediani,** United Arab Emirates University, UAE.
10:30 – 10:45	**Real-Time Speech-Driven Avatar Animation by Predicting Facial landmarks and Deformation Blendshapes** Juan Camilo Vasquez-Correa (VICOMTECH)*; Santiago Moreno-Acevedo (VICOMTECH); Ander Gonzalez-Docasal (VICOMTECH); Aritz Lasarguren (Baleuko); Jone Lòpez (Baleuko); Egoitz Rodriguez (Baleuko); Aitor Álvarez (VICOMTECH)
10:45 – 11:00	**Speech Emotion Recognition for Call Centers using Self-supervised Models: A Complete Pipeline for Industrial Applications** Juan M. Martín-Doñas (Vicomtech)*; Asier López Zorrilla (UPV/EHU); Mikel deVelasco (UPV/EHU); Juan Camilo Vasquez-Correa (VICOMTECH); Aitor Álvarez (Vicomtech); Maria Inés Torres (Universidad del País Vasco UPV/EHU); Paz Delgado (NaturalSpeech); Ane Lazpiur (NaturalSpeech); Blanca Romero (NaturalSpeech); Irati Alkorta (Gureak)
11:00 – 11:15	**GemST: Continual Learning for End-to-End Speech-to-Text Translation** Pranav Karande (VIT University, Vellore)*; Balaram Sarkar (Indian Institute of Technology Indore); Chandresh K Maurya (Indian Institute of Technology Indore)
11:15 – 11:30	**Improved Spoken Emotion Recognition With Combined Segment-Based Processing And Triplet Loss** Dejan Porjazovski (Aalto University)*; Tamas Grosz (Aalto University); Mikko Kurimo (Aalto University)
11:30 – 11:45	**CASCA: Towards the Use of Speaker Role Information in Speaker Diarization** William K Nehrboss (Casca Labs)*

11:45 – 12:00	**Deep Information Maximisation to Mitigate Information Loss in Text Independent Speaker Verification** Nipun Thejan Fonseka (University of Moratuwa)*; Nirmal Sankalana (University of Moratuwa)
12:00 – 12:15	**Improving speaker F0 representation using post-processing techniques** Crisron Rudolf G Lucas (UCD)*; Diptasree Debnath (University College Dublin)
12:15 – 13:30	**Break**
13:30– 15:45	*Oral Session 7: Data and Representation* **Chair: Dr. Koichi Takeuchi**, Okayama University, Japan.
13:30 – 13:45	**FeruzaSpeech: A 60 Hour Uzbek Read Speech Corpus with Punctuation, Casing, and Context** Anna Povey (Redmond High School)*; Katherine Povey (University of Washington)
13:45 – 14:00	**Modeling Score Estimation for Japanese Essays with Generative Pre-trained Transformers** Boago Okgetheng (Okayama University)*; Koichi Takeuchi (Okayama University)
14:00 – 14:15	**PoliTun: Tunisian Political Dataset for Detecting Public Opinions and Categories Orientation** Chayma Fourati (National School of Comupter Science)*; Roua Hammami (Higher Institute of Multimedia Arts); Chiraz Latiri (Faculty of Sciences of Tunis, Tunis EL Manar University); Hatem Haddad (Manouba University)
14:15 – 14 :30	**A Comparison of Different Tokenization Methods for the Georgian Language** Beso Mikaberidze (MIMC)*; Temo Saghinadze (MIMC); Guram Mikaberidze (University of Wyoming); Raphael Kalandadze (Georgian Technical University); Konstantine Pkhakadze (Educational Center for the Georgian Language Technology); Josef van Genabith (Saarland University); Simon Ostermann (German Research Center for Artificial Intelligence); Lonneke van der Plas (Idiap Research Institute); Philipp Müller (DFKI GmbH)
14:30 – 14 :45	**The Qiyas Benchmark: Measuring ChatGPT Mathematical and Language Understanding in Arabic** Shahad Al-Khalifa (King Saud University)*; Hend Al-Khalifa (King Saud University)
14:45 – 15:00	**Data Bias According to Bipol: Men are Naturally Right and It is the Role of Women to Follow Their Lead** Irene Pagliai (University of Göttingen); Goya van Boven (Utrecht University); Oluwatosin Adewumi (Luleå University of Technology)*; Lama Alkhaled (Luleå University of Technology); Namrata Gurung (QualityMinds GmbH); Isabella Södergren (Luleå University of Technology); Elisa H Barney Smith (Luleå Tekniska Universitet)
15:00 – 15:30	**Bulgarian Grammar Error Correction with Data Augmentation and Machine Translation Techniques** Bozhidar Klouchek (The University of Manchester)*; Riza Batista-Navarro (School of Computer Science, The University of Manchester)
15:30 – 15:45	**EEG Signal Analysis for Multimodal Simple Concepts Decoding** Sergio Guillén Jiménez (Universidad de Málaga); Lorenzo J. Tardón (ATIC Research Group. Universidad de Málaga)*; Ana M Barbancho (ATIC Research Group. Universidad de Malaga.); Isabel Barbancho (Universidad de Málaga)
15:45-16:00	*Closing session*

N.B: TIME IN GMT

Presentation duration 12 minutes + 3 minutes for questions.