R2LM 2025

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models

associated with

The 15th International Conference on Recent Advances in Natural Language Processing RANLP'2025

Edited by Alicia Picazo-Izquierdo, Ernesto Luis Estevanell-Valladares, Ruslan Mitkov, Rafael Munoz Guillena and Raul Garcia Cerda

11 September, 2025
Varna, Bulgaria

First Workshop on Comparative Performance Evaluation:
From Rules to Language Models
Associated with the International Conference
Recent Advances in Natural Language Processing
RANLP'2025

PROCEEDINGS

Varna, Bulgaria
11 September 2025

Online ISBN 978-954-452-102-8

Designed by INCOMA Ltd.
Shoumen, BULGARIA

Preface

The rapid growth of deep learning and large language models has transformed the field of natural language processing (NLP), driving remarkable progress in tasks that once seemed out of reach. Yet, as these models scale in size and capability, critical challenges remain. Issues of interpretability, robustness, long-context reasoning, and the substantial data and computational resources required continue to raise questions about their universality. At the same time, rule-based and knowledge-based approaches, once considered traditional or even outdated, are being reconsidered. Their strengths in precision, explainability, and adaptability to low-resource or domain-specific contexts make them valuable complements to data-driven methods. This workshop aims to unite researchers from various fields of symbolic, statistical, and hybrid methods to assess their comparative performance, understand their limitations, and explore how they can work together most effectively. By encouraging dialogue between different research methodologies and highlighting emerging trends such as retrieval-augmented generation, neurosymbolic AI, and knowledge graph-driven systems, we aim to establish a path towards more robust, efficient, and transparent NLP technologies. We hope that the contributions gathered here will not only advance methodological understanding but also inspire a more balanced and inclusive vision for the future of language technology.

Workshop Chairs

- Alicia Picazo-Izquierdo (University of Alicante)
- Ernesto Luis Estevanell-Valladares (University of Alicante, University of Havana)
- Ruslan Mitkov (Lancaster University, University of Alicante)
- Rafael Munoz Guillena (University of Alicante)
- Raul Garcia Cerda (University of Alicante)

Programme Committee

- Constantin Orasan (University of Surrey)
- Paul Greaney (Atlantic Technological University)
- Robiert Sepulveda (University of Alicante)
- Sandra Kuebler (Indiana University)
- Pablo Gervas (Complutense University of Madrid)
- Antonio Toral (University of Alicante)
- Aleksei Dorkin (University of Tartu)
- Sina Ahmadi (University of Zurich)
- Heili Orav (University of Tartu)
- Judith Sieker (Bielefeld University)

Table of Contents

Conference Program

	CoVeGAT: A Hybrid LLM & Graph-Attention Pipeline for Accurate Citation-Aligned Claim Verification Max Bader, Akshatha Arunkumar, Ohan Ahmad, Maruf Hassen, Charles Duong, Vasu Sharma, Sean O'Brien and Kevin Zhu
	A Comparative Study of Vision Transformers and Multimodal Language Models for Violence Detection in Videos Tomas Ditchfield-Ogle and Ruslan Mitkov
16:30–17:00	*Detection of AI-generated Content in Scientific Abstracts* Ernesto Luis Estevanell-Valladares, Alicia Picazo-Izquierdo and Ruslan Mitkov
11:00–11:30	*A Comparative Study of Hyperbole Detection Methods: From Rule-Based Approaches through Deep Learning Models to Large Language Models* Silvia Gargova, Nevena Grigorova and Ruslan Mitkov
11:30–12:00	*Evaluating the Performance of Transformers in Translating Low-Resource Languages through Akkadian* Daniel A. Jones and Ruslan Mitkov
	United We Fine-Tune: Structurally Complementary Datasets for Hope Speech Detection Priya Dharshini Krishnaraj, Tulio Ferreira Leite da Silva, Gonzalo Freijedo Aduna, Samuel Chen, Farah Benamara and Alda Mari
	Does Anaphora Resolution Improve LLM Fine-Tuning for Summarisation? Yi Chun Lo and Ruslan Mitkov
	Transformers and Large Language Models for Hope Speech Detection A Multilingual Approach Diana Patricia Madera-Espíndola, Zoe Caballero-Domínguez, Valeria J. Ramírez-Macías, Sabur Butt and Hector G. Ceballos
12:00–12:30	*Beyond BLEU: Ethical Risks of Misleading Evaluation in Domain-Specific QA with LLMs* Ayoub Nainia, Régine Vignes-Lebbe, Hajar Mousannif and Jihad Zahir
	From Zero to Hero: Building Serbian NER from Rules to LLMs Milica Ikonić Nešić, Sasa Petalinkar, Ranka Stanković and Ruslan Mitkov
	Enhancing the Performance of Spoiler Review Detection by a LLM with Hints Genta Nishi and Einoshin Suzuki
	Evaluating Structured Decoding for Text-to-Table Generation: Evidence from Three Datasets Julian Oestreich and Lydia Müller

17:00–17:30 *Evaluating the LLM and NMT Models in Translating Low-Resourced Languages*
Julita JP Pucinskaite and Ruslan Mitkov

15:00–15:30 *KGEIR: Knowledge Graph-Enhanced Iterative Reasoning for Multi-Hop Question Answering*
Tianda Sun and Dimitar Kazakov

17:30–18:00 *From Handcrafted Features to LLMs: A Comparative Study in Native Language Identification*
Aliyah C. Vanterpool and Katsiaryna Aharodnik

15:30–16:00 *Systematic Evaluation of Rule-Based Analytics for LLM-Driven Graph Data Modelling*
Fabio Antonio Yanez, Andrés Montoyo, Armando Suárez, Alejandro Piad-Morffis and Yudivián Almeida Cruz

Improved Contrastive Learning over Commonsense Knowledge Graphs for Unsupervised Reasoning
Rongwen Zhao and Jeffrey Flanigan

TextBandit: Evaluating Probabilistic Reasoning in LLMs Through Language-Only Decision Tasks
Arjun Damerla, Jimin Lim, Yanxi Jiang, Nam Nguyen Hoai Le and Nikil Selladurai

CoVeGAT: A Hybrid LLM Graph Attention Pipeline for Accurate Citation Aligned Claim Verification

Max Bader[1], **Akshatha Arunkumar**[2], **Ohan Ahmad**[3], **Maruf Hassen**[4], **Charles Duong**[5], **Kevin Zhu**[6]

[1]University of California, Irvine, [2]Monta Vista High School
[3]University at Buffalo, [4]University of Washington
[5,6]Algoverse

mbader@uci.edu, aarunkumar099@student.fuhsd.org
ohanahma@buffalo.edu, marufio@uw.edu

Abstract

In recent years, large language models (LLMs) have demonstrated impressive capabilities in generating human-like textual content. However, their proficiency in accurately verifying quotes and citations remains uncertain. This study benchmarks the effectiveness of contemporary LLMs in assessing the relationship between claims and their cited evidence. To address existing limitations, we propose a novel hybrid approach that integrates multiple verification techniques to robustly evaluate claim-citation alignment.

By systematically combining linguistic parsing, confidence-based semantic verification, and graph neural network modeling, this paper aims to show the enhanced accuracy and interpretability of automated quote and citation verification processing using our method, setting a strong baseline against current LLM capabilities.

1 Introduction

Large language models (LLMs) now draft contracts, summarize court opinions, and tutor students with prose that rivals expert human writing. Yet this fluency masks a structural weakness: current systems freely invent citations, mangle quotations, and misattribute facts (5; 6; 16). Existing "factuality" benchmarks inspect whether a single sentence is plausible, they rarely ask the harder, document-level question, *Does the cited source actually say what the model claims it does?* (9; 11; 3). Consequently, a model can ace popular truthfulness tests while still propagating fabricated evidence (1; 28).

Stop gap fixes remain inadequate. Retrieval-augmented generation merely fetches documents, it does not verify that the retrieved span truly supports the claim (7; 15). Entailment models judge sentence pairs in isolation, ignoring metadata such as author, edition, or publication date (20; 11). Chain-of-thought prompting adds reasoning steps, but those steps themselves can hallucinate, compounding error instead of correcting it. The field therefore, lacks a unified benchmark and methodology that (i) supplies ground-truth claim–evidence pairs, (ii) measures citation alignment end-to-end, and (iii) stresses models with real-world edge cases such as paraphrased quotes, partial attributions, and outdated editions (22; 11).

We address this gap by pairing a meticulously curated dataset with a hybrid verification pipeline. The dataset contains 500 claim–quote pairs drawn from news, legal opinions, scientific papers, and classic literature, each manually labeled for citation correctness. The pipeline chains retrieval, textual entailment, and bibliographic cross-checks into a single decision graph, rejecting any claim unless **all** stages confirm support. Benchmarking GPT-4, Claude 3, Gemini 1.5, Llama 3, and Mistral 7B under this stricter regime reveals that even top models overlook up to 37% of misattributions—failure modes invisible to traditional factuality scores (5; 11).

Our main contributions in this work are as follows:

- **Citation-Alignment Dataset**: a domain-diverse, expert-annotated benchmark focused on whether a quoted span is genuinely present and contextually faithful to its cited source.

- **Hybrid Verification Pipeline**: a modular graph that integrates retrieval, entailment, and

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 1–9, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-001

metadata checks, yielding strict pass-fail judgments rather than scalar plausibility scores.

- **Comprehensive LLM Evaluation**: the first head-to-head comparison of five leading LLM families on citation alignment, uncovering systematic errors that prior metrics miss.

2 Related Work

2.1 Factuality and Hallucination Surveys

Recent work has mapped the 'hallucination' problem, LLMs confidently yielding plausible but unsupported statements, in fine detail. Wang et al. (5) present a comprehensive survey of factuality challenges, grouping failure modes and proposing concrete mitigations. Huang et al. (6) build on this by showing how model scale, decoding strategies, and noisy training data each fuel factual drift. Wang et al. (10) synthesize these findings into a unified framework spanning knowledge extraction, retrieval methods, and domain-specific evaluations. Chen et al. (11) introduce FELM, a long-form factuality benchmark that demonstrates even state-of-the-art evaluators miss subtle inconsistencies. By inspecting each token as it's generated, Barbero et al. (8) catch hallucinations in real time, snaring unsupported fragments before they can snowball. Building on this, Bazarova et al. (14) introduce a topological divergence method for attention graphs, which converts attention weights into topological signatures and rings an alarm whenever the divergence exceeds learned norms, delivering best-in-class detection accuracy and seamless transfer across domains.

2.2 Grounded Citation Methods

Retrieval-augmented generation (RAG) has become the backbone of citation grounding. Thorne et al. (9) established the Fact Extraction and Verification (FEVER) benchmark, pairing claims with supporting Wikipedia passages and setting early standards. Menick et al. (1) then trained GopherCite, a 280B-parameter model, to emit exact inline quotes alongside its answers, reaching 80–90% accuracy on open-domain QA. Huang et al. (6) fine-tuned LLaMA-2-7B to generate line-level citations instead of coarse document IDs, boosting precision by over 14% on the ALCE benchmark. Zhang et al. (7) survey the evolving RAG landscape, while Zhang et al. (12) expand to PolyFEVER, a multilingual, multi-hop testbed. Peng et al. (15) round out this picture by introducing unanswerability checks, ensuring systems gracefully abstain when evidence is lacking.

2.3 Self-Verification

Self-verification routines have emerged to tighten factual accuracy beyond retrieval. Dhuliawala et al. (2) proposed the Chain-of-Verification (CoVe) pipeline: the model drafts an answer, generates check-questions, answers them, and then composes a final response, dramatically reducing unsupported claims. Min et al. (3) introduced FActScore, an automated metric that breaks text into atomic facts and measures support against trusted sources, aligning within 2% of human judgment on biography summaries.

2.4 Quotation Attribution and Multi-Modal Verification

Grounded methods extend beyond factoids to dialogues and multi-modal content. Michel et al. (4) show that LLaMa3 can accurately attribute lines of dialogue to characters across a 28-novel corpus, illustrating how citation techniques translate to narrative text. Recent work by Pang et al. (21) introduces HGTMFC, a hypergraph transformer model that uses fine-grained semantic interactions between text and images for claim verification. This system outperforms prior multi-modal models by using higher-order relationships between textual claims and visual evidence nodes through a hypergraph and line graph propagation. The TREC 2024 RAG Track introduces a citation accuracy benchmark, revealing that LLMs like GPT-4o achieve over 70% alignment with human judgment when verifying grounded citations, even in complex responses (22). However, despite many advancements in factual accuracy, LLMs continue to exhibit significant challenges in generating reliable and accurate citations. Benchmarks compiled by Patel and Anand (28) reveal that even state-of-the-art models often achieve a near-zero accuracy when generating citations, highlighting a critical region for potential research in robust verification.

2.5 Graph-Based and Kernel-Baseline Approaches

Johnson et al. (23) introduce a single, fully shared encoder-decoder neural machine translator model that uses a simple target-language token and a joint subword vocabulary to translate among dozens of languages, achieving state-of-the-art BLEU on major benchmarks, improving low-resource pair performance, and enabling surprisingly effective zero-shot translation by implicitly learning an interlingual representation. Banko et al. (24) build upon the technique of information extraction by employing kernel-based methods and graphical models in order to analyze smaller, domain-specific text to identify and extract pre-defined sets of relationships, laying the groundwork for data-driven linguistic processing. Kriege et al. (25) provide a comprehensive fifteen-year survey of graph-kernel methods, covering neighborhood-aggregation (Weisfeiler-Lehman), assignment-based, substructure, walk-and-path, and attributed-graph approaches. They categorize each technique by feature-extraction paradigm, computational strategy (explicit versus implicit mapping), and support for discrete labels or continuous attributes. Through an extensive empirical study across a variety of datasets, they derive practical guidelines for selecting and tuning graph kernels. More recently, developments in deep learning have extended the usage of graph-based paradigms into advanced graph neural networks (GNNs), using them as powerful tools to analyze non-Euclidean data through interdependencies, helping advance tasks in data mining to natural language understanding by adapting principles in the graph structures of deep learning (26). Within the development of NMT specifically, recent advancements have been shown with the integration of GNNs, in particular the multi-level community awareness graph neural network (MC-GNN) proposed by Nguyen et al. (27), which can explicitly model composite semantics like morphology, syntax, and complex linguistic information by leveraging graph structures, sometimes substituting components to enhance the quality of translation.

2.6 Gaps and Our Contribution

Despite its strengths, our CoVeGAT introduces a novel citation verification pipeline that combines dependency-based SVO extraction with graph attention mechanisms, outperforming traditional classifiers on benchmark datasets. However, several key limitations remain. First, the pipeline depends heavily on the accuracy of SVO extraction; parsing errors, especially in idiomatic or complex constructions, cascade through the entire system. Second, our CoVeGAT assumes claims can be fully decomposed into discrete triplets, which overlooks temporal reasoning, multi-sentence context, and implicit premises that our sliding-window backup cannot capture. Third, the dense semantic graphs required for each citation pair can be computationally expensive to construct at scale. Finally, CoVeGAT's performance hinges on access to high-quality, domain-specific labeled data for fine-tuning the graph attention model, limiting its generalizability across disciplines. Future work may explore integrating neural semantic parsers, lightweight graph construction methods, or few-shot adaptation strategies to address these constraints and extend CoVeGAT's applicability to real-world, low-resource domains.

3 Methodology

Our overall goal is to take unstructured text, namely, free-form claims paired with their supporting citations, and convert it into a graph-structured dataset that explicitly records which triplets are supported or contradicted by the citation. This allows downstream models to reason about which pieces of a claim hold up against evidence and which do not. To achieve this, we have developed a fully automated dataset construction pipeline (See Figure 1), comprising four sequential stages.

By the end of this pipeline, every claim-citation pair is represented as a small graph whose nodes and edges are richly tagged with support scores, forming a large, trainable dataset for any model that needs to reason over evidence.

3.1 Triplet Extraction

We utilize the spaCy NLP library (17) to perform semantic parsing on both claims and their corresponding citation texts. Each complex sentence is simplified into structured Subject-Verb-Object (SVO) triplets, capturing fundamental semantic relationships. This process explicitly captures negation within verbs by prefixing negated verbs with "NOT_". The decomposition of these sentences helps reduce textual complexity and enables focused comparisons between claim and citation content.

If no clear SVO triplets are extracted using this dependency parsing, our method defaults to a sliding-window trigram approach, similar in spirit to open information extraction (18; 24). This ensures robust extraction even from short or less well-structured texts. Our multi-tiered approach to parsing effectively distills complex sentences into fun-

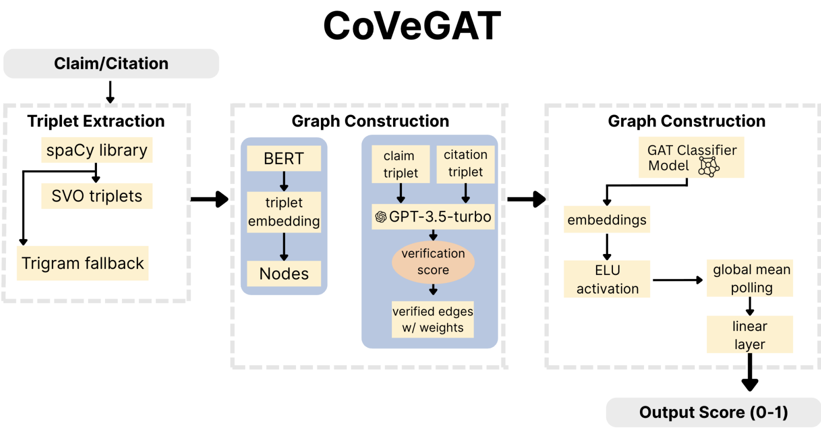


Figure 1: Overview of the CoVeGAT architecture. First, claim–citation pairs are passed through an SVO-based triplet extractor (with a trigram fallback) to produce semantic subject–verb–object nodes, whose embeddings are obtained via BERT. Edges between claim and citation triplets are weighted by verification scores produced by GPT-3.5-turbo. The resulting weighted graph is then fed into a graph attention classifier (GAT), with ELU activations, global mean pooling, and a final linear layer to produce a normalized output score in [0, 1].

damental semantic relationships, facilitating precise comparisons between claim and citation.

3.2 Chain-Of-Verification (CoVe)

To be able to assess the evidential support provided by the citations accurately, CoVe utilizes an external model, simulated via OpenAI's GPT-3.5-turbo. Each extracted triplet from a claim is evaluated against the citation text, which results in confidence scores ranging from 0 to 1. Scores closer to 1 indicate higher confidence and stronger evidential support, while scores closer to 0 indicate low confidence and weak or no evidential support. This reflects the likelihood of semantic entailment. These scores serve as quantifiable measures of evidential strength between individual triplets, extending the Chain-of-Verification framework (2) and aligning with recent work evaluating LLMs as factuality judges (16).

3.3 Graph Construction

We construct a weighted semantic graph by representing claim and citation triplets as nodes. Edges between these nodes are established based on CoVe-derived confidence scores (2), which effectively encode the strength of evidential relationships as edge weights. This graph captures the nuanced semantic dependencies and interactions between claim statements and their potential evidential references, enabling downstream reasoning through graph attention mechanisms (19).

3.4 Graph Attention Network (GAT) Analysis

The final stage of this process involves analyzing the constructed graph using a graph attention network (GAT) (19). This neural network architecture leverages node features, derived from BERT embeddings of triplet components (39), and weighted edges in order to aggregate semantic information. The GAT model specifically pools information from claim-side nodes to make graph-level classifications, ultimately determining whether a claim is supported by its citation.

By integrating semantic parsing, confidence-based verification, and advanced graph neural networks, CoVeGAT provides an interpretable approach to automated quote and citation verification.

4 Experimental Methodology

4.1 Dataset

Source. Our experiments use AVeriTeC—a 4 568-claim benchmark for real-world fact verification that aggregates checks from 50 independent organisations. From the official release, we draw exactly 500 claims from the dev.json split, retaining only the raw claim texts and their ground-truth verdicts. The dev partition is preferred because it is entirely disjoint from the training data supplied with the dataset, ensuring our evaluation corpus is unseen by any baseline that might have been pre-trained on the original training split.

To create a balanced testbed, we generate a one-to-one set of 500 fabricated counterparts. Each fabricated claim is derived from its real twin by

Model	Label accuracy	Macro-F1	Abstain rate
Perplexity 70 B	28.2 %	43.4 %	71.7 %
GPT-4o	72.2 %	76.2 %	17.7 %
Gemini 1.5 Pro	82.5 %	86.3 %	10.8 %
DeepSeek-MoE 67 B	69.7 %	80.1 %	30.3 %
Copilot-Turbo	76.4 %	82.4 %	19.1 %
Claude 3 Opus	44.3 %	57.2 %	55.7 %
Mistral-7B-Instr.	81.4 %	87.0 %	15.4 %

Table 1: Model performance on classification task

applying a single, controlled perturbation chosen uniformly at random:

- Named-entity substitution (e.g., swapping "Angela Merkel")
- Numerical alteration (changing dates, counts, or magnitude)
- Temporal shift (advancing or back-dating events)
- Causal inversion (reversing cause and effect clauses)

All edits are automated by the Python script provided in our code repository and manually spot-checked to eliminate obvious lexical cues that would trivialise classification.

The procedure yields a 1,000-item dataset with a perfectly balanced label distribution: 500 accurate and 500 inaccurate statements.

4.2 Evaluation

Evaluation Metrics. We report three standard measures:

- Label Accuracy (LA) – the fraction of quotes whose predicted label exactly matches the gold 3-way label set (Accurate / Inaccurate / Cannot Determine).
- Macro-F1 – the unweighted F1 average over the two decisive classes (Accurate and Inaccurate); any Cannot Determine output is treated as an error. This balances precision and recall and is insensitive to the 50 / 50 class split.
- Abstain Rate – the percentage of quotes that a model marks Cannot Determine, included because several LLMs prefer to hedge rather than commit.

For the non-parametric CoVe-Kernel baseline, we also log the raw kernel-score distribution and the hit rate at the empirical decision cutoff $\tau = 0.025$ (see Implementation section).

Baselines. We benchmark seven large-language models plus one embedding-based system:

- Perplexity 70B (PPL-70B) (31) – Commercial MoE model accessed via the Perplexity AI chat API.
- GPT-4o (36) – OpenAI's flagship model (June 2025 weights).
- Gemini 1.5 Pro (37) – Google Gemini; abstains least often (108 "cannot-determine" decisions in our run).
- DeepSeek-MoE 67B (32) – Chinese–English mixture-of-experts model.
- GitHub Copilot Turbo (33) – GPT-4-Turbo derivative served in Copilot Chat.
- Claude 3 Opus (34) – Anthropic's top-tier model; most cautious, highest abstain rate.
- Mistral 7B-Instruct (35) – Open-weights model queried through the HuggingFace Inference API, included to gauge how a freely available 7B model fares.
- CoVe-Kernel – Our reproduction of Chain-of-Verification: MiniLM embeddings (38), RBF kernel, $\tau = 0.025 \rightarrow$ "Accurate" if the claim–evidence distance is below the threshold, "Inaccurate" if above, and "Cannot Determine" in a ±0.002 band around τ.

All LLMs are evaluated zero-shot. Each receives batches of 25 quotes with the fixed prompt:

> "For each numbered statement, reply on its own line with one of:
>
> Accurate and true | Inaccurate and false | Cannot determine.
>
> Be specific in your evaluation and rely on trustworthy sources when possible."

Decoding temperature is 0.0, and responses are capped at four tokens per quote to prevent extra commentary.

Refer to Table 1 for the complete results.

5 Results

5.1 Overall Performance

On the mixed dataset of 1,000 shuffled quotes (500 authentic, 500 fabricated), Google Gemini 1.5 Pro achieves the highest raw accuracy (82.5 %) while the open-weights Mistral-7B-Instruct posts the best balanced score (87.0 % macro-F1). GPT-4o follows at 72.2 %, its accuracy held back by a habit of replying, cannot determine about one claim in six.

Models that abstain heavily lose ground: Claude 3 Opus and Perplexity 70 B hedge on more than half of the inputs and finish below the 50 % line despite respectable precision on the items they do judge.

The results exhibit a clear trend. With identical prompts and deterministic decoding, models that frequently answer Cannot Determine (i.e., adopt a cautious strategy) suffer lower overall accuracy, whereas more decisive systems—such as Gemini 1.5 Pro and LLaMA-2-7B-Instruct—achieve higher scores, albeit at the cost of occasional confident errors on fine-grained numeric edits. Model size alone is not the primary determinant of performance; with well-designed instruction tuning, a 7-billion-parameter model can match, and in certain metrics surpass, commercial systems in the 70–100 billion-parameter range.

5.2 Methodology performance

We also ran a non-parametric CoVe-Kernel check on the 500-item set supplied. Each row contains an RBF similarity score between a quote and its evidence; by convention, a score below 0.025 is taken to mean "the quote is false" (i.e. CoVe thinks it has spotted a factual mismatch). Under that single rule the system flags 482 of 500 quotes correctly, an accuracy of 96.4 %, leaving only 18 errors.

All 18 mistakes lie inside a very narrow band just above the threshold (0.025 – 0.035). Inspection shows three recurring causes:

1. Tiny numeric edits. Changing "42 million" to "41 million" shifts only one token and barely moves the embedding, nudging the score above τ even though the meaning flips.

2. Entity swaps with extra framing. Sentences like "It is widely believed that Theresa May ..." add hedging phrases the original lacked; the additional words expand vector distance enough to miss the cutoff.

3. Causal inversions hidden in long sentences. When "X led to Y" becomes "Y led to X" inside a 30-word clause, most tokens stay identical, and cosine distance again changes only marginally.

Because every error sits within 0.010 of the boundary, simply lowering τ to a score such as 0.022 would raise recall on false claims without creating many false positives; but it would also erase any chance of labelling a quote true. The underlying limitation is that MiniLM embeddings are too coarse-grained for subtle factual reversals; swapping the encoder for a task-tuned cross-encoder or introducing a small margin band (Cannot Determine for 0.023–0.027) are straightforward ways to harden the system.

In short, with a hand-picked threshold CoVe-Kernel can spot blatant fabrications with high precision, but it remains brittle around fine-grained numeric or causal tweaks—exactly the corner cases that modern LLMs also find most challenging.

6 Discussion

Our evaluation of eight citation-verifying systems, including several advanced LLMs and one hybrid non-parametric method, reveals key trends about the strengths and limitations of current approaches to automated claim citation verification. The results demonstrate that while LLMs have made progress in factual reasoning, their ability to judge claim-evidence alignment consistently remains uneven, especially in adversarial or subtly perturbed contexts.

6.1 Performance vs. Prudence Tradeoff

A clear pattern emerges in the relationship between decisiveness and performance. Models like Gemini 1.5 Pro and Mistral-7B-Instruct, which issue definitive judgments with relatively low abstention rates (10.8% and 15.4%, respectively), achieve the highest overall accuracy and macro-F1 scores. In contrast, Claude 3 Opus and Perplexity 70B adopt a cautious stance, abstaining from over half the inputs, underperforming on both precision weighted and overall correctness. This emphasizes a central challenge in ethical LLM deployment: overly conservative models risk failing to flag misinformation, while confident ones may propagate falsehoods when it does not reflect factual correctness.

Furthermore, model size was not the primary determinant of performance. Despite having fewer parameters, Mistral-7B-Instruct outperformed several larger commercial systems, highlighting the value of instruction tuning and alignment strategies over raw scale. This suggests that accessible, open weight models, when carefully tuned, can achieve advanced performance in citation-sensitive tasks without requiring proprietary infrastructure.

6.2 Fine-Grained Factuality Remains Elusive

Both LLMs and the CoVe-Kernel method struggled with subtle perturbations, especially numeric alterations and causal inversions. In contrast, the CoVe-Kernel system achieved 96.4% accuracy on its benchmark, with every error clustered near the decision threshold, revealing a sensitivity to edge cases. Such failure modes emphasize that vector distance, while capturing semantic similarity, is insufficient for ensuring factual equivalence. In practical terms, changing "42 million" to "41 million" or flipping cause-effect relationships produced only minor shifts in embedding space, small enough to evade detection by both LLMs and shallow similarity functions, highlighting a need for deeper analysis beyond word overlap in critical domains like journalism and legal review.

6.3 Ethical Implications and Design Considerations

Our findings carry several implications for the design and deployment of LLMs in citation-sensitive environments. First, models that over-rely on confidence or refuse to abstain when uncertain about data may contribute to hallucinated factuality, the illusion of truth created by authoritative tone and plausible structure. Second, the tendency of some models to abstain excessively raises the risk of ethical ambiguity, failing to identify misinformation when a judgment is expected.

The high performance of a relatively simple CoVe-Kernel baseline further raises questions about the interpretability and transparency of LLM outputs. Unlike most LLMs, which offer little insight into why a given citation was judged as accurate, the kernel-based method provides direct access to distance thresholds and can be calibrated to balance precision and recall. This suggests that hybrid systems, like our CoVE-Kernel system, may offer a more robust path forward for citation verification.

7 Conclusion

This study evaluated whether state-of-the-art LLMs can reliably distinguish true statements from minimally perturbed fabrications. We constructed a 1,000-item test set by pairing 500 verified AVeriTeC claims with single-edit counterparts, each manually validated to remove superficial cues. Seven zero-shot LLMs and a CoVe-Kernel baseline were assessed using label accuracy, macro-F1, and abstention rate.

Decisive models like Google Gemini 1.5 Pro (82.5 % accuracy) and Mistral-7B Instruct (87.0 % macro-F1) consistently outperformed cautious systems such as Claude 3 Opus and Perplexity 70 B, which abstained on over half of the inputs and fell below 50 % overall accuracy. The CoVe-Kernel approach, relying on MiniLM embeddings with a single RBF cutoff, achieved 96.4 % accuracy, underscoring the competitiveness of simple, interpretable methods.

These results reveal a pronounced trade-off between decisiveness and restraint: lower abstention rates drive higher accuracy, whereas excessive hedging imposes substantial performance costs. Crucially, model scale alone does not determine success; instruction tuning and calibrated abstention thresholds are equally decisive.

Future work should (1) enhance small encoders or cross-encoders to detect subtle numeric and causal perturbations and (2) develop fully integrated pipelines that unify fine-grained citation ("sanitation"), systematic self-verification ("verification"), and atomic evaluation metrics such as FActScore. Such end-to-end frameworks promise to advance the reliability and transparency of LLM-based fact-verification systems.

8 References

References

[1] Menick, J.; Kadav, A.; Jaques, N.; Chen, M.; Petrov, M.; Hesse, C.; Clark, C. Teaching Language

Models to Support Answers with Verified Quotes. *arXiv:2203.11147*, 2022.

[2] Dhuliawala, S.; Min, S.; Zhan, C.; Narayan-Chen, T.; Yasunaga, M.; McCann, B.; Prabhakaran, V. Self-Verification Improves Few-Shot Reasoning. *arXiv:2305.14251*, 2023.

[3] Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.-t.; Koh, P. W.; Iyyer, M.; Zettlemoyer, L.; Hajishirzi, H. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. 2023.

[4] Michel, G.; Epure, E. V.; Hennequin, R.; Cerisara, C. Evaluating LLMs for Quotation Attribution in Literary Texts: A Case Study of LLaMa3. 2024.

[5] Wang, Y.; Wang, M.; Manzoor, M. A.; Liu, F.; Georgiev, G.; Das, R. J.; Nakov, P. Factuality of Large Language Models: A Survey. In *Proceedings of EMNLP 2024*, 2024.

[6] Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; Liu, T. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *arXiv:2311.05232*, 2023.

[7] Zhang, Y.; Liu, S.; Qin, Z.; Wan, X.; Feng, Y. Evaluation of Retrieval-Augmented Generation: A Survey. *arXiv:2405.07437*, 2024.

[8] Barbero, A.; Carvalho, J.; Bode, N.; West, A.; Peterson, J. Robust Hallucination Detection in LLMs via Adaptive Token Selection. *arXiv:2504.07861*, 2025.

[9] Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Mittal, A. FEVER: a Large-scale Dataset for Fact Extraction and Verification. In *EMNLP*, 2018.

[10] Wang, C.; Liu, X.; Yue, Y.; Tang, X.; Zhang, T.; Cheng, J.; Yao, Y.; Gao, W.; Hu, X.; Qi, Z.; Wang, Y.; Yang, L.; Wang, J.; Xie, X.; Zhang, Z.; Zhang, Y. Survey on Factuality in Large Language Models: Knowledge, Retrieval and Domain-Specificity. *arXiv:2310.07521*, 2023.

[11] Chen, S.; Zhao, Y.; Zhang, J.; Chern, I.-C.; Gao, S.; Liu, P.; He, J. FELM: Benchmarking Factuality Evaluation of Large Language Models. In *NeurIPS Workshops*, 2023.

[12] Zhang, H.; Anjum, S.; Fan, H.; Zheng, W.; Huang, Y.; Feng, Y. Poly-FEVER: A Multilingual Fact Verification Benchmark for Hallucination Detection in LLMs. *arXiv:2503.16541*, 2025.

[13] Ma, H.; Xu, W.; Wei, Y.; Chen, L.; Wang, L.; Liu, Q.; Wu, S.; Wang, L. EX-FEVER: A Dataset for Multi-hop Explainable Fact Verification. In *Findings of ACL*, pp. 9340–9349, 2024.

[14] Bazarova, A.; Yugay, A.; Shulga, A.; Ermilova, A.; Volodichev, A.; Polev, K.; Belikova, J.; Parchiev, R.; Simakov, D.; Savchenko, M.; Savchenko, A.; Barannikov, S.; Zaytsev, A. Hallucination Detection in LLMs with Topological Divergence on Attention Graphs. 2025.

[15] Peng, et al. Unanswerability Evaluation for Retrieval Augmented Generation. 2024.

[16] Fu, X.-Y.; Laskar, M. T. R.; Chen, C.; Tn, S. B. Are Large Language Models Reliable Judges? A Study on the Factuality Evaluation Capabilities of LLMs. In *GEM Workshop at NeurIPS*, pp. 310–316, 2023.

[17] Honnibal, M.; Montani, I. spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. *TACL*, 5, 2017.

[18] Mausam; Schmitz, M.; Soderland, S.; Bart, R.; Etzioni, O. Open Language Learning for Information Extraction. In *EMNLP-CoNLL*, 2012.

[19] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In *ICLR*, 2018.

[20] Feher, D.; Khered, A.; Zhang, H.; Batista-Navarro, R.; Schlegel, V. Learning to Generate and Evaluate Fact-Checking Explanations with Transformers. *arXiv:2410.15669*, 2024.

[21] Pang, H.; Li, C.; Zhang, L.; Wang, S.; Zhang, X. Beyond Text: Fine-Grained Multi-Modal Fact Verification With Hypergraph Transformers. In *AAAI*, vol. 39, pp. 6389–639, 2025.

[22] Thakur, N.; Pradeep, R.; Upadhyay, S.; Campos, D.; Craswell, N.; Lin, J. Support Evaluation for the TREC 20 4 RAG Track: Comparing Human versus LLM Judges. *arXiv:2504.15205*, 2025.

[23] Johnson, M.; Schuster, M.; Thorat, N.; Krikun, M.; Wu, Y.; Chen, Z.; Viégas, F.; Wattenberg, M.; Corrado, G.; Hughes, M.; Dean, J. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *TACL*, 5, pp. 339–351, 2017. DOI:10.1162/tacl_a_00065

[24] Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; Etzioni, O. Open Information Extraction from the Web. In *IJCAI*, p . 26 0–2676, 2007.

[25] Kriege, N. M.; Johansson, F. D.; Giscard, P.-L. A Survey on Graph Kernels. *arXiv:1903.11836*, 2019.

[26] Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P. S. A Comprehensive Survey on Graph Neural Networks. *IEEE TNNLS*, 32(1), pp. 4–24, 2021. DOI:10.1109/TNNLS.2020.2978386

[27] Nguyen, B.; Nguyen, L.; Dinh, D. Multi-level Community-awareness Graph Neural Networks for Neural Machine Translation. In *COLING*, pp. 5021–5028, 2022.

[28] Patel, M.; Anand, A. Factuality or Fiction? Benchmarking Modern LLMs on Ambiguous QA with Citations. *arXiv:2412.18051*, 2024.

[29] Tonmoy, S. M. I.; Zaman, S. M. M.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; Das, A. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. *arXiv:2401.01313*, 2024.

[30] Wang, Y.; Wang, M.; Manzoor, M. A.; Liu, F.; Georgiev, G.; Das, R. J.; Nakov, P. Factuality of Large Language Models: A Survey. *arXiv:2402.02420*, 2024.

[31] Perplexity AI. *Perplexity AI Assistant* [computer software]; https://www.perplexity.ai/ (accessed 2025-08-29).

[32] DeepSeek AI. *DeepSeek-MoE 67B* [computer software]; https://huggingface.co/deepseek-ai/deepseek-67b-base (accessed 2025-08-29).

[33] GitHub. *GitHub Copilot Turbo* [computer software]; https://github.com/features/copilot (accessed 2025-08-29).

[34] Anthropic. *Claude 3 Opus* [computer software]; https://www.anthropic.com/news/claude-3-family (accessed 2025-08-29).

[35] Mistral AI. *Mistral 7B-Instruct* [computer software]; https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2 (accessed 2025-08-29).

[36] OpenAI. *GPT-4o* [computer software]; https://platform.openai.com/docs/models/gpt-4o (accessed 2025-08-29).

[37] Google. *Gemini 1.5 Pro* [computer software]; https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/1-5-pro (accessed 2025-08-29).

[38] Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *NeurIPS*, 2020.

[39] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 2019.

A Comparative Study of Vision Transformers and Multimodal Language Models for Violence Detection in Videos

Tomas Ditchfield-Ogle
School of Computing and Communications
Lancaster University
Lancaster, United Kingdom
tomas.o.ogle@gmail.com

Ruslan Mitkov
University of Alicante
ruslan.mitkov@ua.es

Abstract

This study compares methods for detecting violent videos, which are crucial for ensuring real-time safety in surveillance and digital moderation. It evaluates four approaches: a random forest classifier, a transformer model, and two multimodal vision-language models. The process involves preprocessing datasets, training models, and assessing accuracy, interpretability, scalability, and real-time suitability. Results show that traditional methods are simple but less effective. The transformer model achieved high accuracy, and the multimodal models offered high violence recall with descriptive justifications. The study highlights trade-offs and provides practical insights for the deployment of automated violence detection.

1 Introduction

Concerns about harmful content have prompted the UK government to implement the Online Safety Act 2023, which encourages proactive content moderation and violence prevention both online and offline (GOV UK Department for Science, 2025). As smart cities evolve, citizens demand enhanced safety measures and swift emergency responses, pressuring authorities to adopt automation tools (Pujol et al., 2020). Governments are facing the rapid growth of video content in surveillance and digital applications, making manual analysis impractical. This drives the need for real-time video systems that identify patterns for safety and emergencies (Sabha and Selwal, 2024). Social media platforms also struggle to manage vast video volumes in near real-time (Pujol et al., 2020), amid increasing circulation of violent content, including hate crimes and terrorist attacks (Studer, 2017). Many platforms, such as Facebook and YouTube, attempt to moderate content through automated tools; however, the scale and immediacy of live streaming make this nearly impossible (Pujol et al., 2020).

Automatic violence detection is difficult due to its inherent subjectivity. Violent acts are not always visually explicit and depend on context, like body posture, group dynamics, or weapons, posing barriers to definition (Naik and Gopalakrishna, 2017). Other issues include illumination variance, which affects outdoor video quality due to changes in lighting, such as day/night transitions or weather, impacting colour and contrast (Kaur and Singh, 2024).

Fortunately, AI offers promising tools, particularly through computer vision (CV) and machine learning models trained to classify visual data. This project explores and compares four such methods for detecting violence: Random Forest classifier, TimeSformer, Llama 3.2

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 10–20, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-002

Vision Instruct and Janus Pro. This project aims to investigate the effectiveness of cutting-edge machine learning technologies in detecting real-world violence. This research has applications ranging from enhancing online moderation to improving street safety in cities.

2 Background

Recent research indicates that machine learning models are increasingly supporting video classification, particularly in the context of violence detection (VD).

2.1 Violence detection (VD) software

Before Deep Learning (DL) methods, VD was seen as recognising specific human actions (Peixoto et al., 2020). Following initial approaches, DL techniques improved VD results (Peixoto et al., 2020), notably with 3D convolutional neural networks for extracting spatio-temporal patterns (Ding et al., 2014). However, many models remain computationally intensive, often using multi-stream input and stacked LSTM layers, with limited details on their complexity (Ullah et al., 2019) (Ullah et al., 2022). Some researchers focus on models balancing performance and efficiency. One achieved 87.25% accuracy with just 0.27 million parameters (Cheng et al., 2021a) using depth-wise separable convolutions from Pseudo-3D Residual Networks (Qiu et al., 2017) and MobileNet (Howard et al., 2017), thereby reducing complexity without compromising accuracy. The VD field has evolved from handcrafted features to advanced DL models that interpret video spatio-temporal cues.

2.2 Random Forest

Random Forests are a popular machine learning model used for classification and forecasting, requiring high-quality data for training. They improve algorithms and user behaviour analysis, aiding pattern recognition (Salman et al., 2024). The model excels in classification and regression, using cross-validation for accuracy and handling missing data effectively (Achari and Sugumar, 2024). It also reduces bias by training multiple decision trees on random subsets of data, making it one of the most reliable techniques in ML (Salman et al., 2024). Random forests are often used in hybrid models for VD. For example, a study developed a facial recognition assault system using Random Forest, achieving 98% precision and 97.5% accuracy, showing ensemble methods enhance safety (Ohwosoro et al., 2024).

2.3 TimeSformer

Video understanding tasks, such as VD, require models to interpret spatial and temporal features. TimeSformers, which utilise a transformer-based architecture with spatio-temporal attention, reason across frames and time (Bertasius et al., 2021a). Research has found TimeSformer performs well in DeepFake detection (Chen et al., 2024). The architecture is suitable for VD, where an extended temporal context is key. TimeSformer differs from standard Transformers in that it learns spatio-temporal features directly from frame patches. Research shows that "divided attention," applying temporal and spatial attention separately, achieves the highest video classification accuracy (Bertasius et al., 2021b).

2.4 Large Language Models

Following ChatGPT's launch, attention has focused on large language models (LLMs) (Tian et al., 2024), especially for their strong performance in classification (Al Faraby et al., 2024), summarisation (Doss et al., 2024), data and code generation (Shimabucoro et al., 2024) (Nejjar et al., 2025). The rapid development of LLMS is clear in the late 2023 and early 2024 releases of Google's Gemini, Anthropic's

Claude 3, and OpenAI's GPT-4 (Shahriar et al., 2024). These models represent a significant leap in capabilities, transitioning from text-only to multimodal understanding across text, images, and audio, with enhanced parameters and speed (Shahriar et al., 2024). LLMs' ability to understand and generate extensive data has created opportunities, such as Llama3.2, which addresses predatory conversations and abusive messages (Arora, 2025). However, these models can gain vision capabilities. Vision in LLMs means adapting transformer models from language to interpret images (Yenduri et al., 2024). This has expanded the generative pre-trained transformer (GPT) to include vision. Since multimodal LLMs are relatively new, many research areas are still in their early stages (Wang et al., 2024). OpenAI's GPT-4 release in May 2024 marked a key shift, as it was the first to interpret emotions from videos (Islam and Moushi, 2024), opening up new applications. Yet, using multimodal LLMs for VD in videos remains under-explored, with research gaps this project aims to fill (Jaafar and Lachiri, 2023).

3 Data

The violent samples in both datasets depict real-world street fight scenarios recorded under varying conditions. Non-violent samples include everyday activities like walking, eating, and playing sports, representing a wide range of non-aggressive behaviours. This diversity provides a realistic setting for evaluating safety monitoring and automated incident detection systems.

3.1 Ethical concerns

All datasets used in this study were obtained from publicly available academic sources. No new data was collected, annotated, or shared during the project. The RLVS dataset was sourced from Kaggle (Mustafa, 2020) and introduced initially by Soliman et al. (Soliman et al., 2019). The RWF-2000 dataset was downloaded from Hugging Face and first presented by Cheng et al. (Cheng et al., 2021b). Both datasets contain publicly available videos designed for violence detection research. None of the content includes personally identifiable information, as all videos were either anonymised or publicly accessible.

3.2 RLVS Dataset

A subset of the Real-Life Violence Situations (RLVS) dataset comprising 957 violent samples and 839 non-violent samples was used for training, validation, and in-distribution testing. As a result, the final dataset used in this study. A consistent train/validation/test split was generated and saved in a persistent JSON file to support reproducibility.

Figure 1: Example frames from RLVS. Left: violent, Right: non-violent

3.3 RWF-2000 Dataset

The RWF-2000 dataset was used solely for out-of-distribution testing to assess generalisation beyond the training data. A subset of 383 violent and 395 non-violent was used for testing to reduce computational load, especially for the vision-language models. Models were neither trained nor fine-tuned on RWF-2000, and no manual labelling or editing was done.

4 Methodology

4.1 Data Preprocessing

A unified pipeline ensured consistent inputs across models. Videos were uniformly sampled every 15 frames, with up to 16 frames per clip. Frames were resized to 224×224; clips with fewer than 8 valid frames were discarded. Training and validation sets were augmented with brightness, contrast, and saturation shifts (±20%), hue shifts (±0.1), horizontal flips, random crops, and rotations (±10°), applied consistently across frames to preserve temporal coherence. RLVS was split via stratified sampling (20% test, 10% validation); for RWF-2000, a subset was used for testing only. Motion features were derived from Farneback optical flow, summarised with statistics (mean, variance, skewness, range, etc.) and a high-motion pixel count. Frames were saved as JPEGs (for LLMs) and as tensors ($T \times C \times H \times W$) in PyTorch format for efficient loading.

4.2 Random Forest

To establish a classical baseline, a Random Forest classifier was trained on motion features derived from dense optical flow.

4.2.1 Training

The Random Forest model was trained on motion statistics derived from dense optical flow, including mean, median, standard deviation, maximum, minimum, range, skewness, variance, and the proportion of high-motion pixels per frame. These features capture both overall motion intensity and its distribution across frames. Labels were assigned automatically from dataset filename prefixes ('violent-' or 'nonviolent-'), consistent with the dataset's original annotation scheme. Hyperparameters were optimised via grid search with five-fold cross-validation, using ROC-AUC as the scoring metric due to class imbalance. The best model employed 100 estimators, a maximum depth of 8, a minimum sample split of 10, a minimum sample leaf of 4, and balanced class weighting. This configuration was retrained on the whole training set and evaluated on the test set.

4.2.2 Feature Importance

Post-training, feature importances revealed that mean and minimum motion magnitudes were the most influential predictors, underscoring the role of motion intensity in distinguishing between violent and non-violent activity.

4.2.3 Interpretability of Random Forest

The Random Forest provides insight into which features drive classification. For example, a high mean motion magnitude strongly predicted violent sequences, such as street fights, whereas a low minimum flow magnitude aligned with stable, non-violent scenes. However, the model also produced false positives in contexts like crowd surges at sports events, where collective movement mimicked aggression. These results suggest that RF's interpretability is valuable, but its rule-like motion thresholds are not robust across diverse scenarios.

4.3 TimeSformer

To establish a DL benchmark, a transformer-based video classification model was implemented using the TimeSformer architecture. TimeSformer builds on the Vision Transformer (ViT) framework by introducing a mechanism to handle both spatial and temporal dimensions in video data. Rather than using traditional 3D convolutions, it applies divided attention across space and time separately, enabling efficient and scalable video understanding from raw pixel data.

4.3.1 Model Configuration

The TimeSformer model used was `facebook/TimeSformer-base-finetuned-k400`, pre-trained on the Kinetics-400 dataset (Bertasius et al., 2021a). To adapt it for binary violence detection, the classification head was replaced with a fully connected layer producing two logits. Frames were converted to floating point and normalised by dividing by 255.0, preserving dynamic range without distorting pixel intensities. Temporal tensors were zero-padded as needed. Labels were inferred from filename prefixes. The entire model was fine-tuned to adapt specifically to the task.

4.3.2 Training Configuration

Training used the Hugging Face Trainer API with a batch size of 6. The model was optimised for cross-entropy loss with label smoothing (0.1), a learning rate of 5e-5, and cosine scheduling, along with a 25% warm-up. Early stopping had a patience of 2 steps. Regularisation included weight decay (0.2) and gradient clipping (norm 1.0). Evaluation occurred every 1000 steps, saving the best model based on validation log loss. Seeds were fixed at 42 for reproducibility. Training ran on a SLURM job with an NVIDIA A5000 GPU.

4.4 Interpreting TimeSformer decisions

We analyse the model's posterior $p(\text{violence} \mid x)$ without binarisation. To expose its decision process, we extract self-attention from each transformer block during a forward pass. (`output_attentions=true`) and apply attention rollout: heads are averaged, an identity residual added, rows normalised, and attention matrices multiplied across layers to form a single CLS→patch relevance map. This is reshaped into a $g \times g$ grid and temporal tubelets to yield spatial heatmaps (time-averaged). In the Kinetics-400 TimeSformer, inputs are 224×224 with 16×16 patches ($g = 14$) and tubelet size 8, so 16 frames give $T_{\text{eff}} = 2$ temporal tokens inferred at run time. For windowed videos, video-level probabilities are aggregated with monotone poolers that preserve probabilistic semantics: $\max$, mean, top-k mean, log-sum-exp (temperature τ), and noisy-OR $1 - \prod_i (1 - p_i)$. Evaluation utilities proper scoring rules (negative log-likelihood, Brier) and calibration/ranking metrics (ROC–AUC, PR–AUC, ECE/MCE).

4.5 Multimodal LLMs: Llama 3.2 Vision and Janus-Pro-7B

To assess the potential of multimodal large language models for violence detection, Meta's Llama 3.2 Vision Instruct model and DeepSeek's Janus-Pro-7B were used.

4.5.1 Llama 3.2 Vision Instruct Model Configuration

The model (`meta-Llama/Llama-3.2-11B-Vision-Instruct`) was loaded via Hugging Face Transformers (meta, 2024) with mixed-precision evaluation. Inputs were processed using AutoProcessor for image normalisation and prompt tokenisation. Generation was limited to 200 tokens with deterministic decoding (do_sample=False, temperature=0.2) to ensure stable outputs. Each frame was evaluated using the following prompt: "This image is part of a public dataset of street and public scenes used for academic research. Start your response with a yes or no if violence is depicted in this image. Then describe what is happening. If a violent or aggressive incident occurs, describe what happened and identify those involved. If there isn't any violence, describe the scene as peaceful or non-violent. Use simple language and avoid complex terms where possible."

4.5.2 Janus-Pro-7B Vision Model Configuration

Janus was loaded using AutoModelForCausalLM with mixed precision enabled. The Janus-specific VLChatProcessor was used to process images and chat-style prompts, ensuring consistent resizing, normalisation, and tokenisation. The prompt used was identical to that used with Llama. Generation parameters were configured with do_sample=False, repetition_penalty=1.0, and a maximum of 200 new tokens to produce deterministic and focused outputs.

4.6 Testing Methodology

To evaluate model effectiveness and generalisability, two testing settings were used: in-domain testing on the RLVS test split and out-of-domain testing on a subset of RWF-2000. This allowed assessment of performance within the original data distribution and in unseen environments. A unified preprocessing and evaluation pipeline standardised video extraction, transformation, and organisation for both datasets. The RLVS test set consisted of 363 videos (194 violent and 169 non-violent), which were held out from training and validation. The RWF-2000 subset included 778 videos (383 violent, 395 non-violent), enabling fair cross-model comparison. To improve robustness and simulate real-world variability, data augmentation was applied at the video level with a 50% probability during RLVS training and validation. Extracted frames were then formatted as inputs for the three models. For input preparation, the Random Forest model used optical flow between consecutive frames to extract nine motion statistics forming fixed-length feature vectors. The TimeSformer model received RGB frame tensors of shape (16, 3, 224, 224) and applied spatial and temporal self-attention for classification. Llama and Janus processed frames individually with a fixed prompt; a zero-shot text classifier classified their generated text outputs to assign violent or non-violent labels. Outputs were compared to ground truth labels, with confusion matrices used to analyse false positives and negatives. Performance metrics included accuracy, precision, recall, and F1-score.

5 Evaluation

5.1 In-Domain Testing (RLVS)

Model	Accuracy	Precision	Recall	F1	Time (s)
Random Forest	77.96%	0.7614	0.8556	0.8058	0.01
TimeSformer	96.41%	0.9547	0.9793	0.9669	39.90
LLaMA	78.24%	0.7154	0.9845	0.8286	99 309.80
Janus Pro	74.38%	0.6772	0.9948	0.8058	9 904.08

Table 1: In-domain RLVS test set performance.

Table 1 summarises model performance on the RLVS test set. TimeSformer performed strongest, achieving high accuracy and a balanced precision–recall trade-off with inference times suitable for near-real-time surveillance. LLaMA and Janus Pro reached very high recall, but this came at the cost of precision, often misclassifying non-violent group behaviour as violent. Random Forest was the fastest model, classifying samples almost instantly; however, its reliance on simple motion statistics made it prone to errors in ambiguous scenarios, such as crowd surges. These results suggest that TimeSformer is best suited for automated monitoring, while multimodal models may be more valuable in forensic review or moderation contexts where interpretability is prioritised. Random Forest, despite weaker performance, remains attractive for highly constrained deployments. The error distributions for each RLVS model are illustrated in the corresponding confusion matrices (Figure 2), which make explicit the balance between false positives and false negatives discussed above.

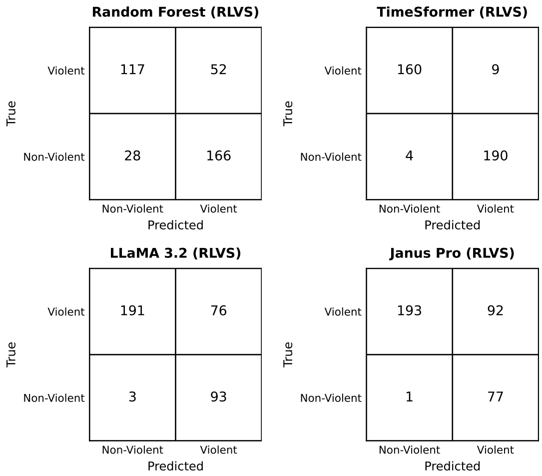


Figure 2: Confusion matrices on the RLVS test set (rows = true labels, columns = predicted labels).

5.2 Out-Of-Domain Testing (RWF-2000)

Model	Accuracy	Precision	Recall	F1	Time (s)
Random Forest	54.24%	0.5754	0.2689	0.3665	0.01
TimeSformer	68.76%	0.6590	0.7571	0.7047	82.16
LLaMA	64.78%	0.5873	0.9635	0.7298	193 729
Janus Pro	74.68%	0.6691	0.9635	0.7898	21 706

Table 2: Out-of-domain RWF-2000 test set performance.

Table 2 presents the performance of all models on the RWF-2000 dataset. Janus Pro achieved the highest F1 score (0.79) with near-perfect recall (0.96), demonstrating strong zero-shot transfer capabilities. LLaMA achieved similar recall but with lower precision, resulting in a higher number of false positives. Both models generated interpretable outputs, though their runtimes were extremely high. TimeSformer generalised well, despite being fine-tuned only on RLVS, achieving balanced scores and completing inference in just over a minute. Random Forest performed poorly under distribution shift, with low recall and F1, reflecting limited robustness. Overall, Janus Pro showed the strongest zero-shot generalisation, while TimeSformer offered a better balance of speed and accuracy. LLaMA remained interpretable, but it was computationally intensive. Random Forest remained the most efficient but least adaptable. Error patterns under distribution shift are shown in the RWF-2000 confusion matrices (Figure 3), which emphasise the models' differing capacities to generalise.

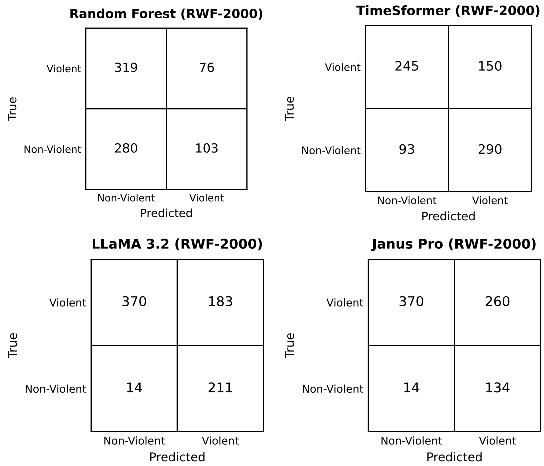


Figure 3: Confusion matrices on the RWF-2000 test set (rows = true labels, columns = predicted labels).

5.3 Decision evidence and probability quality

On 778 test windows (383 violent; 395 non-violent), the model yields ROC–AUC 0.767 and PR–AUC 0.764 from raw posteriors. Probability quality is moderate (negative log-likelihood 1.94; Brier 0.288) and calibration indicates over-confidence (ECE 0.286, 15 bins). Cumulative gains show that the top 10% of windows by $p(\text{violence} \mid x)$ contain 18.3% of violent windows (Lift@10% 1.83). Aggregating windows improves video-level ranking: *noisy OR* reaches ROC–AUC 0.800 (PR–AUC 0.751), while *top-k mean* ($k = 3$) gives the best proper scoring (negative log-likelihood 1.851; Brier 0.287) and the lowest ECE among the poolers.

Pooler	ROC–AUC ↑	NLL ↓
max	0.787	2.011
mean	0.755	1.854
noisy–OR	**0.800**	2.149
log–sum–exp	0.759	1.864
top-k mean	0.761	**1.851**

Table 3: Video-level pooling of window probabilities (no thresholds). Best per column in bold.

5.4 Qualitative evidence

Spatial overlays for high-confidence violent windows focus on converging bodies and limbs, with temporal peaks in the tubelet that captures contact. Low confidence violent windows show diffuse attention, often in pre- or post-event frames, under occlusion, or when brief actions are split across tubelets. High confidence non-violent windows emphasise crowd surges or celebratory gestures that are visually salient yet non-violent. Figure 4 shows examples.

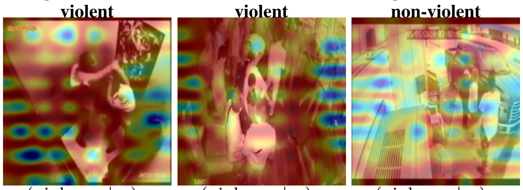


Figure 4: Spatial attention overlays (CLS→patch relevance) for three representative windows.

5.5 Evaluation Findings

5.5.1 Notable Observations

TimeSformer consistently achieved the best balance of precision and recall across both datasets. Its confusion matrices indicated lower rates of false positives and false negatives, supporting its reliability in varied scenarios. LLaMA Vision exhibited a strong bias towards recall, detecting violent content aggressively but occasionally misclassifying benign scenes as violent. This trade-off may be acceptable in high-sensitivity contexts but is less suitable where false positives carry a significant cost. Random Forest performed reliably on RLVS, particularly in identifying non-violent scenes, but its accuracy declined on RWF-2000. This shows its handcrafted features generalise poorly to more varied or noisy data.

5.5.2 Error Analysis and Interpretability

TimeSformer's errors primarily resulted from crowded or celebratory scenes, which produced false positives, and short, low-contrast violent clips, which resulted in false negatives. LLaMA and Janus Pro often hallucinated aggression, labelling cricket games as violent. Random Forest struggled with camera shake and noisy backgrounds, exposing its reliance on clean motion signals. Interpretability also varied. TimeSformer's attention maps highlighted human interactions, usually aligning with the source of violence. LLaMA and Janus Pro generated natural language explanations, offering detailed scene descriptions of actors, actions, and context, such as environments and expressions. These outputs exposed systematic biases and helped diagnose false positives. They also added value in human-in-the-loop scenarios where moderators could review justifications alongside predictions.

5.5.3 Summary of Findings

TimeSformer delivered the highest overall performance, with strong generalisation and the fewest false positives. Its ability to model spatial and temporal features makes it well-suited for continuous, high-precision surveillance in environments where alert reliability is critical.

LLaMA Vision and Janus Pro achieved the highest recall, demonstrating strong sensitivity to violent content and producing interpretable natural language explanations. These qualities make them valuable for content moderation and investigative or regulatory settings, where comprehensive flagging and explanation are prioritised. However, their lower precision and very high inference times limit their suitability for real-time or autonomous applications.

Random Forest, while fast and transparent, generalised poorly to RWF-2000. Its simplicity and efficiency still make it viable for controlled edge deployments, such as low-power CCTV units, where latency and interpretability take precedence over accuracy. Overall, these findings emphasise distinct deployment niches: TimeSformer as the most balanced and scalable solution, multimodal LLMs for human-in-the-loop systems, and Random Forest for resource-constrained contexts. Together, they illustrate the trade-offs between accuracy, interpretability, and efficiency that must guide real-world adoption of violence detection systems.

6 Conclusion

6.1 Project Limitations

This study has several limitations. The models were not trained to detect weapons, as this was not included in the datasets used in this project, which limits their ability to detect armed violence. No post-hoc calibration was applied, ensuring fairness across models but potentially constraining accuracy and generalisability. Attention maps serve as explanatory aids rather than causal attributions but consistently emphasise physical interaction.

6.2 Future Work

Future work should evaluate these models in real-time surveillance or moderation settings. Adding audio cues, such as raised voices, could support earlier detection. Another approach is to incorporate textual commentary from speech transcripts or subtitles, as verbal threats often precede violence. LLMs can process text and video jointly, enabling cross-modal reasoning. In contrast, models like TimeSformer would need auxiliary NLP components or architectural changes. Methodological steps include aligning subtitles with video frames, fine-tuning multimodal encoders, and comparing late-fusion against joint-embedding approaches to determine which best captures temporal and semantic dependencies. Such integration could provide richer context and improve robustness in safety-critical applications.

6.3 Summary

This project compared four approaches to AVD in video: a Random Forest baseline, the transformer-based TimeSformer, LLaMA 3.2 Vision Instruct and Janus Pro, evaluated on RLVS and RWF-2000 datasets. TimeSformer achieved the strongest balance of accuracy and efficiency, making it suitable for real-world deployment. LLaMA Vision demonstrated high recall and interpretability, which is valuable in settings with human oversight; however, computational demands limit its scalability. The Random Forest was lightweight and interpretable but struggled to generalise, highlighting the limits of handcrafted features. Overall, transformer-based models appear most promising when balancing performance and scalability. Future directions include model distillation, real-time optimisation, and audio integration.

Acknowledgements

This work has been partially supported by the CIDEXG/2023/12 project, funded by the Generalitat Valenciana. We thank Dexter Latcham and Suzannah Read for their reading of earlier drafts and editorial suggestions; any remaining errors are our own.

References

A Prudhvi Sai Kumar Achari and R Sugumar. 2024. Performance analysis and determination of accuracy using machine learning techniques for naive bayes and random forest. In *AIP Conference Proceedings*, volume 3193, page 020199. AIP Publishing LLC.

Said Al Faraby, Ade Romadhony, et al. 2024. Analysis of llms for educational question classification and generation. *Computers and Education: Artificial Intelligence*, 7:100298.

Ankush Arora. 2025. *Detecting Online Abuse: Fine-Tuning LLMs for Abusive Language Detection*. Ph.D. thesis, Universität Koblenz.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021a. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, pages 813–824. PMLR.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021b. Is space-time attention all you need for video understanding?

Zhengxuan Chen, Shuo Wang, Deyang Yan, and Yushi Li. 2024. A spatio-temporl deepfake video detection method based on timesformer-cnn. In *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pages 1–6. IEEE.

Ming Cheng, Kunjing Cai, and Ming Li. 2021a. Rwf-2000: An open large scale video database for violence detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4183–4190. IEEE.

Ming Cheng, Kunjing Cai, and Ming Li. 2021b. Rwf-2000: an open large scale video database for violence detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4183–4190. IEEE.

Chunhui Ding, Shouke Fan, Ming Zhu, Weiguo Feng, and Baozhi Jia. 2014. Violence detection in video by using 3d convolutional neural networks. In *Advances in Visual Computing: 10th International Symposium, ISVC 2014, Las Vegas, NV, USA, December 8-10, 2014, Proceedings, Part II 10*, pages 551–558. Springer.

Srinath Doss et al. 2024. Comparative analysis of news articles summarization using llms. In *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, pages 1–6. IEEE.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

Noussaiba Jaafar and Zied Lachiri. 2023. Multimodal fusion methods with deep neural networks and meta-information for aggression detection in surveillance. *Expert Systems with Applications*, 211:118523.

Gurmeet Kaur and Sarbjeet Singh. 2024. Revisiting vision-based violence detection in videos: A critical analysis. *Neurocomputing*, 597:128113.

meta. 2024. [link].

Mohamed Mustafa. 2020. Real-Life Violence Situations Dataset. `https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset`.

Anuja Jana Naik and MT Gopalakrishna. 2017. Violence detection in surveillance video-a survey. *International Journal of Latest Research in Engineering and Technology (IJLRET)*, 1:1–17.

Mohamed Nejjar, Luca Zacharias, Fabian Stiehle, and Ingo Weber. 2025. Llms for science: Usage for code generation and data analysis. *Journal of Software: Evolution and Process*, 37(1):e2723.

ID Ohwosoro, AE Edje, and CO Ogeh. 2024. A hybrid assault detection system using random forest enabled xgboost-lightgbm technique. *Nigerian Journal of Science and Environment*, 22(2):177–189.

Bruno Peixoto, Bahram Lavi, Paolo Bestagini, Zanoni Dias, and Anderson Rocha. 2020. Multimodal violence detection in videos. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2957–2961. IEEE.

Francisco A Pujol, Higinio Mora, and Maria Luisa Pertegal. 2020. A soft computing approach to violence detection in social media for smart cities. *Soft Computing*, 24(15):11007–11017.

Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541.

Ambreen Sabha and Arvind Selwal. 2024. Towards machine vision-based video analysis in smart cities: a survey, framework, applications and open issues. *Multimedia Tools and Applications*, 83(22):62107–62158.

Hasan Ahmed Salman, Ali Kalakech, and Amani Steiti. 2024. Random forest algorithm overview. *Babylonian Journal of Machine Learning*, 2024:69–79.

Innovation & Technology GOV UK Department for Science. 2025. Online safety act: Explainer.

Sakib Shahriar, Brady D Lund, Nishith Reddy Mannuru, Muhammad Arbab Arshad, Kadhim Hayawi, Ravi Varma Kumar Bevara, Aashrith Mannuru, and Laiba Batool. 2024. Putting gpt-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency. *Applied Sciences*, 14(17):7782.

Luísa Shimabucoro, Sebastian Ruder, Julia Kreutzer, Marzieh Fadaee, and Sara Hooker. 2024. Llm see, llm do: Guiding data generation to target non-differentiable objectives. *arXiv preprint arXiv:2407.01490*.

M. Soliman, M. Kamal, M. Nashed, Y. Mostafa, B. Chawky, and D. Khattab. 2019. Violence recognition from videos using deep learning techniques. In *Proceedings of the 9th International Conference on Intelligent Computing and Information Systems (ICICIS'19)*, pages 79–84, Cairo, Egypt.

Grace Studer. 2017. Live streaming violence over social media: an ethical dilemma. *Charleston L. Rev.*, 11:621.

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C Comeau, et al. 2024. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *Briefings in Bioinformatics*, 25(1):bbad493.

Fath U Min Ullah, Mohammad S Obaidat, Khan Muhammad, Amin Ullah, Sung Wook Baik, Fabio Cuzzolin, Joel JPC Rodrigues, and Victor Hugo C de Albuquerque. 2022. An intelligent system for complex violence pattern analysis and detection. *International journal of intelligent systems*, 37(12):10400–10422.

Fath U Min Ullah, Amin Ullah, Khan Muhammad, Ijaz Ul Haq, and Sung Wook Baik. 2019. Violence detection using spatiotemporal features with 3d convolutional neural network. *Sensors*, 19(11):2472.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. 2024. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*.

Gokul Yenduri, M Ramalingam, G Chemmalar Selvi, Y Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G Deepti Raj, Rutvij H Jhaveri, B Prabadevi, Weizheng Wang, et al. 2024. Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*.

Detection of AI-generated Content in Scientific Abstracts

Ernesto Luis Estevanell-Valladares
University of Havana
University of Alicante
ernesto.estevanell@ua.es
ernesto.estevanell@matcom.uh.cu

Alicia Picazo-Izquierdo
University of Alicante
alicia.picazo@ua.es

Ruslan Mitkov
Lancaster University
University of Alicante
r.mitkov@lancaster.ac.uk

Abstract

The growing use of generative AI in academic writing raises urgent questions about authorship and the integrity of scientific communication. This study addresses the detection of AI-generated scientific abstracts by constructing a temporally anchored dataset of paired abstracts—each with a human-written version that contains scientific abstracts of works published before 2021 and a synthetic version generated using GPT-4.1. We evaluate three approaches to authorship classification: zero-shot large language models (LLMs), fine-tuned encoder-based transformers, and traditional machine learning classifiers. Results show that LLMs perform near chance level, while a LoRA-fine-tuned DistilBERT and a PassiveAggressive classifier achieve near-perfect performance. These findings suggest that shallow lexical or stylistic patterns still differentiate human and AI writing, and that supervised learning is key to capturing these signals.

1 Rationale

The proliferation of generative artificial intelligence (AI) models, particularly large language models (LLMs), has significantly reshaped content creation across domains (Kreps et al., 2022), including scientific writing. While these models offer powerful tools for drafting, summarising, and translating academic texts, their capacity to autonomously generate scientific abstracts raises ethical concerns regarding authorship, originality, and the integrity of scholarly communication. In the context of peer-reviewed publication, the need of distinguishing between human-written and AI-generated content is becoming increasingly pressing. Without reliable detection methods, academic institutions, publishers, and reviewers face the risk of unknowingly legitimising AI-generated content, undermining trust in the scholarly record. As such, there is an urgent need for robust tools capable of accurately identifying AI-generated scientific writing, particularly in the early, high-stakes stages of academic dissemination—namely, paper abstracts.

Several recent approaches have emerged to address this challenge. Tools such as OpenAI's AI Text Classifier and GPTZero have attempted to leverage statistical and linguistic features to differentiate AI from human writing, with varying levels of success. In parallel, research studies have investigated stylometric patterns, perplexity metrics, and discourse-level anomalies as potential indicators of synthetic text. However, most of these efforts suffer from limitations including small or general-domain datasets, lack of temporal anchoring (e.g., comparing texts written before the advent of LLMs), and insufficient validation on high-quality, domain-specific academic corpora. Consequently, there remains substantial room for advancement in this area.

Our study seeks to address the above gaps by constructing a temporally controlled and domain-specific corpus for AI writing detection in scientific abstracts. By compiling a set of abstracts published prior to 2021—before the rise of transformer-based language models—and juxtaposing them with a parallel set of abstracts generated by state-of-the-art LLMs for the same papers, we aim to compare different models to distinguish between human and AI-generated scientific writing. This approach not only ensures a clear temporal boundary between human-authored and synthetic texts but also contributes a novel, curated dataset to the field of natural language processing.

The remainder of this paper is structured as follows: Section 2 reviews related work on AI text detection and scientific authorship analysis. Section 3 details the construction of the human and synthetic abstract corpora. Section 4 outlines our model architecture and experimental setup, as well as the results obtained. Finally, Section 5 discusses

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 21–29, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-003

the implications of our findings and Section 6 includes conclusions as well as directions for future research.

2 Related Work

The growing use of generative artificial intelligence (GAI), particularly large language models (LLMs), in scientific writing has inspired a broad spectrum of academic research. Recent research explores customisation strategies, potential pitfalls, and the promising capabilities of these tools in scholarly contexts. This section covers the use of generative AI in scientific writing and highlights different state-of-the-art methods for detecting AI-generated content.

2.1 AI-generated scientific writing

Emerging research highlights the ways in which commercial AI systems are being adapted for scientific use. Some studies compare multiple AI chatbots to demonstrate performance across academic writing tasks, with GPT-4 scoring highest in quantitative assessments, though all models failed to produce original scientific contributions (Lozić and Štular, 2023). Similarly, Biondi-Zoccai et al. (2025) provide a detailed overview of AI tools tailored for manuscript drafting, refinement, and literature review. While tools like ChatGPT, Grammarly, and SciSpace Copilot are becoming increasingly embedded in academic workflows, the authors caution against their uncritical adoption. In a practical example, Babl and Babl (2023) test ChatGPT's capacity to generate a conference abstract from fictitious data. The output, despite minor hallucination in the references, was structurally sound and content-appropriate, raising concerns over undetectable AI involvement in academic submissions.

A major concern addressed in the literature is the issue of hallucinations—false or fabricated information produced by AI. (Athaluri et al., 2023) (2023) thoroughly examine this phenomenon in scientific writing, warning of its potential to mislead readers and reviewers and to contaminate academic discourse. Another critical analysis comes from Jenko et al. (2024), who evaluate AI-generated literature reviews in musculoskeletal radiology. The study reveals significant factual inaccuracies and shallow content, concluding that current AI tools cannot yet replace expert domain knowledge in scientific synthesis. These risks are echoed in (Biondi-Zoccai et al., 2025), who warn of AI's susceptibility to generating fraudulent datasets and paper mill content. Traditional plagiarism detectors are ineffective against this sophisticated output, calling for robust AI detection mechanisms.

Despite these issues, several sources underscore the potential benefits of AI-assisted writing. (Huang and Tan, 2023) (2023) describe how ChatGPT can improve review article composition by accelerating literature organisation, enhancing linguistic clarity, and assisting non-native English speakers. They argue that AI serves best as a co-authoring assistant—providing structural and linguistic support while the scientist retains control over content and critical interpretation.

2.2 Detection of AI-generated content

As large language models (LLMs) such as GPT-4o and DeepSeek become capable of producing highly coherent and human-like text across multiple domains and languages, researchers have responded by developing diverse strategies and platforms to identify machine-generated content. These approaches generally fall into three categories: traditional machine learning, transformer-based detection models, and zero-shot evaluations using state-of-the-art LLMs themselves.

Early efforts in AI text detection relied heavily on traditional machine learning models using surface-level linguistic features (Alghamdi et al., 2023); (Jawahar et al., 2020). These include metrics such as token diversity, sentence length distributions, part-of-speech frequencies, and syntactic patterns. Classifiers such as Support Vector Machines (SVMs), trained on engineered features extracted from labelled datasets, have demonstrated moderate success. However, with the rise of transformer-based architectures, detection strategies have increasingly moved toward fine-tuned pretrained language models. Fine-tuning models such as BERT and DeBERTa-v3 on domain-specific corpora, often with techniques like Low-Rank Adaptation (LoRA), have shown improved performance (Hans et al., 2024);(He et al., 2021). A third, more recent direction involves evaluating the ability of advanced LLMs to detect AI-generated content in a zero-shot setting (Papageorgiou et al., 2024); (Forment et al., 2025). This strategy leverages the generative model itself—such as GPT-4o-mini—to assess whether a given text appears AI-generated.

Benchmark datasets have played a crucial role in driving these developments. Notable resources include the AuTexTification corpus (used in IberLEF 2023 and 2024), GPT-2 Output Dataset, HC3 and HC3 Plus for chat-based detection (Su et al., 2024), and domain-specific sets like TweepFake (Fagni et al., 2021) and MGTBench (He et al., 2024). These corpora span a range of languages, modalities, and genres—offering fertile ground for cross-domain benchmarking.

Detection tasks have also become the focus of organised evaluation campaigns. Shared tasks such as the IberLEF AuTexTification challenge, the SemEval 2024 Task 8 on authorship verification, and upcoming initiatives at RANLP and COLING have galvanised research efforts by offering competitive benchmarks and standardised test sets. These tasks increasingly emphasise multilingualism and domain diversity, reflecting real-world challenges where generative AI is used in both high-resource and under-resourced linguistic settings.

All in all, current detection platforms rely on a spectrum of techniques, from transparent ML classifiers (Alghamdi et al., 2023) to opaque but powerful deep learning systems (Hashmi et al., 2024); (Mahmud et al., 2024). Despite incremental gains in accuracy, no approach currently guarantees robust, generalisable detection across domains, languages, and use cases. The limitations of zero-shot LLM detection and the rising fluency of AI outputs all point to the need for hybrid approaches and labelled datasets.

3 Dataset

This project focuses on the development of a structured, balanced, and semantically coherent dataset designed to support research on the automatic identification and classification of machine-generated versus human-written scientific abstracts. In order to evaluate this task with high fidelity and domain diversity, we compiled a dataset that not only spans a wide range of scientific disciplines but also ensures that each data point includes two corresponding versions of the same abstract: one written by a human and another generated by a machine.

The entire data pipeline—from initial collection to the final preparation of train and test sets—was carefully engineered to respect the semantic integrity of abstract pairs and the thematic proportionality of the dataset. This section outlines the key stages of that process, namely the dataset compilation via API scraping and metadata filtering, followed by a custom train-test split procedure that guarantees class balance, category proportionality, and the preservation of human-machine abstract pairs.

3.1 Original and generated abstracts

The human-written abstracts were collected leveraging the Semantic Scholar Graph API to retrieve metadata and abstracts for a wide range of scientific papers across multiple disciplines. The query process was domain-driven, using keywords and filters to target articles in areas such as medicine, physics, environmental science, engineering, computer science, chemistry, biology, and materials science.

For each query result, the script extracted several fields of interest, including the paper's title, abstract, year of publication, venue, DOI, unique paper ID, and URL. Additional metadata was collected when available through integrations with the Unpaywall and Crossref APIs, which were used to verify open-access status and ensure the retrievability of the original documents.

To maintain linguistic and disciplinary consistency, the script applied a series of filtering criteria. First, only abstracts written in English were retained, as determined using the langdetect library. A minimum abstract length threshold was enforced to guarantee sufficient content for accurate language detection. Second, the script discarded non-research content, such as editorials or metadata-only entries, and prioritised papers for which a PDF was accessible or openly licensed. A human curation and review process was also implemented to verify abstract consistency and validity.

Once cleaned and filtered, each abstract was stored along with its associated metadata in a structured format. These abstracts constitute the human-authored portion of the final dataset.

The machine-generated abstracts were produced using the model GPT-4.1. For each scientific article retrieved in the previous stage, the first 10 pages of the full-text document were used as input to the model. These pages were either extracted from the available PDFs or obtained through additional metadata queries and processing pipelines that reconstructed the document's main body content.

The GPT-4.1 model was prompted to generate an abstract that closely followed the conventions of scientific abstract writing: summarising the re-

Domain	Human	Machine
biology	13,057	12,512
business	7,047	6,763
chemistry	11,770	10,838
computer science	9,163	8,361
economics	6,410	6,943
education	14,503	10,028
engineering	12,044	11,140
environmental science	18,313	13,229
materials science	8,574	7,740
medicine	20,063	17,782
physics	28,910	26,160
sociology	7,413	6,249

Table 1: Total word count per category for human- and machine-written abstracts.

search problem, methodology, and key findings in a concise and coherent format. No abstract was generated unless a minimum threshold of source content was available (i.e., a full 10-page span or an equivalent amount of text). This ensured that the machine-generated abstract had sufficient context and detail to mirror the function and structure of the original human-written abstract.

All generated abstracts were paired with their corresponding human-written versions using the paper's title as a unique ID, and both versions shared the same category and metadata. This pairing process resulted in a clean and balanced dataset where each title appears exactly twice—once under the label human and once under the label machine.

The total word count analysis reveals consistent patterns across categories, with human-written abstracts generally containing slightly more words than their machine-generated counterparts. This trend is observed in nearly all disciplines, most notably in fields like medicine, physics, and environmental science, which show the highest overall word volumes. The discrepancy in length may reflect differences in content density, verbosity, or summarisation strategies between human authors and the language model.

3.2 Split with pair integrity

Once the full dataset of human–machine abstract pairs had been compiled and validated, the next step was to divide it into a training set and a test set, in a way that would enable reliable supervised learning and fair evaluation. This division was carried out with particular attention to three key

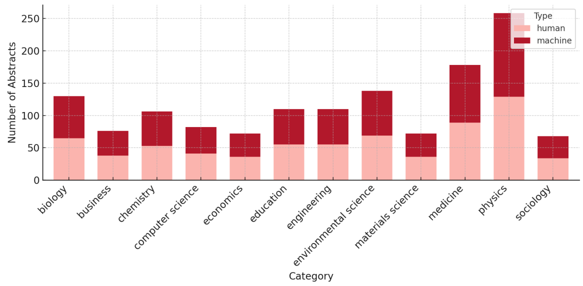


Figure 1: Train split

requirements: semantic pairing integrity, class balance, and thematic proportionality across scientific categories.

The core structural unit of the dataset is the abstract pair, consisting of one human-written and one machine-generated version of the same scientific paper. In order to prevent data leakage and preserve the semantic boundary between training and test samples, it was essential that these pairs remain intact during the split. That is, both the human and machine versions of a given abstract had to be assigned to the same subset—either training or test. Splitting the two across subsets would have introduced significant risk of semantic overlap, as both versions are derived from the same source paper and often convey similar core content.

To enforce this constraint, the split was performed at the level of the paper title, which uniquely identifies each pair. Only titles that appeared exactly twice in the dataset—once with each version—were eligible for inclusion. The total pool of such valid pairs was then randomly divided into training and test sets using an 80/20 stratified split, with stratification based on the category assigned to each paper. This ensured that the topical distribution of abstracts across disciplines (e.g., physics, medicine, computer science) remained proportionally balanced in both subsets.

After assigning titles to either the training or test set, all associated abstracts and metadata were recovered using the title as the join key. This approach guaranteed that the final training and test sets were (i) fully balanced in terms of class labels (human and machine); (ii) proportionally distributed across scientific categories (iii) free from any leakage or overlap of semantically equivalent texts.

Following the train–test split, a final validation step was performed to ensure the integrity of the abstract pair structure within each subset. This

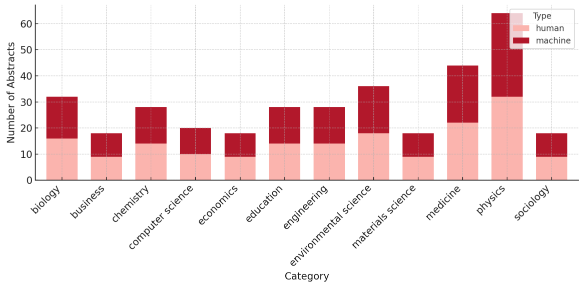


Figure 2: Test split

involved verifying that each paper ID appeared exactly once per version (i.e., once for the human-written abstract and once for the machine-generated one), and that both instances were assigned to the same subset.

This was achieved by counting the frequency of each ID in the training and test sets independently. The results of this verification confirmed that all pairs were preserved and correctly assigned, with no instances of cross-subset leakage or structural inconsistencies.

4 Experiments

This study investigates the capacity of computational models to classify scientific abstracts according to their authorship—human or machine-generated. The classification task was designed as a binary decision problem and explored through three complementary modelling approaches: (1) prompt-based classification using large language models (LLMs), (2) supervised fine-tuning of a transformer-based classifier with parameter-efficient adaptation, and (3) traditional machine learning pipelines based on bag-of-words representations.

4.1 Experimental setup

In this section we aim to explore which model and configuration performs best when classifying human vs. machine generated text. To this end, different setups have been explored and are detailed below.

Prompt-based LLM classification In the first setup, a suite of instruction-tuned large language models (LLMs) was used to perform zero-shot classification. Each model was prompted with a research abstract and asked to determine whether it had been written by a human or generated by a machine. A fixed prompt template was used for all models to ensure consistency and comparability across predictions.

- System prompt: You are a diligent assistant that labels research abstracts. Reply strictly with either 'human' or 'machine' and nothing else.
- User prompt: Classify the following abstract as written by a human or by a machine. Answer with only 'human' or 'machine'. Abstract: * Classification: *

No few-shot examples were provided, and no additional formatting was required from the model output beyond the binary label. Different models with different parameter configuration and size were used:

- OpenAI: GPT-4.1, o4-mini, GPT-4o-mini
- LLaMa 4: llama4-scout-instruct-basic, lama4-maverick-instruct-basic
- Qwen3: qwen3-30b-a3b, qwen3-235b-a22b
- DeepSeek: deepseek-r1-basic

Fine-tuned transformer with LoRA To complement zero-shot inference with supervised learning, we employed the AutoGOAL AutoML framework (Estevez-Velarde et al., 2020) to automatically explore and optimise deep learning pipelines based on transformer architectures. AutoGOAL was extended to include 44 pipeline variants across 13 transformer-based language models (introduced by Estevanell-Valladares et al., 2024), sourced from the Hugging Face model hub (Jain, 2022). These models included various fine-tuning strategies: full fine-tuning, partial fine-tuning (top-layer adaptation), and Low-Rank Adaptation (LoRA).

Training and evaluation were performed on a workstation equipped with an NVIDIA RTX 4090 GPU, allowing efficient gradient-based learning across configurations. Each pipeline was evaluated using 2-fold stratified cross-validation on the training set. The best-performing pipeline selected by AutoGOAL used LoRA fine-tuning over a DistilBERT base model.

Traditional machine learning baseline To establish a non-neural baseline, we also constructed and tuned a traditional machine learning pipeline built on sparse vector representations. The pipeline consisted of a HashingVectorizer for text featurisation and a PassiveAggressiveClassifier for classification.

The HashingVectorizer was configured to use over two million features, binary encoding, and L1 normalisation, transforming text into a fixed-length sparse binary representation. The classifier was optimised with a high aggressiveness parameter (C=9.991) and evaluated using stratified validation on the training set.

4.2 Results

The performance of the large language models (LLMs) on the binary classification task—determining whether a scientific abstract was written by a human or generated by an AI—revealed a consistent trend: despite strong general-purpose capabilities, the models exhibited difficulty distinguishing between the two classes in a reliable manner.

Across all LLMs evaluated, F1 scores remained low, rarely exceeding 0.34. The best-performing model, Qwen3-235B, achieved an F1 score of 0.335, followed closely by GPT-4.1 and DeepSeek-R1, with scores of 0.333 and 0.332 respectively. Accuracy scores hovered near 49–50% for most models, suggesting that predictions were often close to chance level in aggregate, despite marginal gains in class-specific precision or recall. The confusion matrix in Figure 3 suggests that Qwen3-235B, which is the best LLM, almost always mistakes every machine-generated abstract for human writing.

This performance gap highlights a critical limitation of general-purpose LLMs when applied to subtle authorship attribution tasks involving highly similar content, such as pairs of human- and machine-written scientific abstracts derived from the same paper. The task appears to require more fine-grained discriminative capabilities than current zero-shot prompting strategies afford.

In contrast, the best-performing model emerged from a supervised approach using LoRA fine-tuning on top of the distilbert-base-multilingual-cased encoder. This configuration, discovered through AutoGOAL's AutoML pipeline search, achieved a markedly superior F1 score of 0.974, with equivalent levels of accuracy, precision, and recall. These results underscore the value of task-specific training, particularly when using parameter-efficient fine-tuning techniques like LoRA.

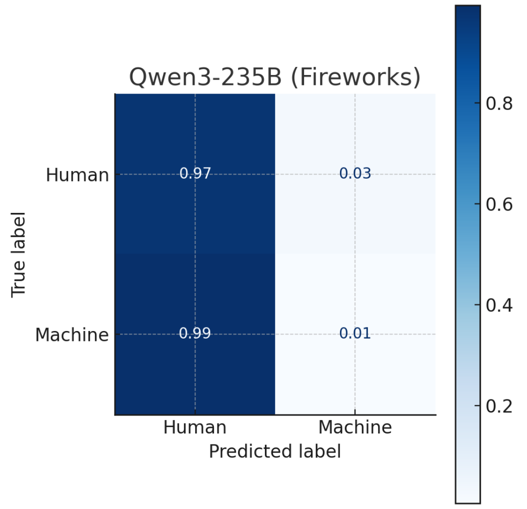


Figure 3: Confussion matrix for the best-performing LLM

Model	Acc	P	R	F1
LoRA DistilBERT	**0.974**	**0.974**	**0.974**	**0.974**
PassiveAggressive	0.972	0.972	0.972	0.972
Qwen3-235B	0.490	0.357	0.490	0.335
GPT-4.1	0.487	0.328	0.487	0.333
DeepSeek-R1	0.494	0.280	0.493	0.332

Table 2: Accuracy, precision, recall, and F1 score of the best-performing models across the classification approaches.

The fine-tuned encoder demonstrated consistent and robust performance across all metrics, correctly classifying nearly all abstracts in the test set. This outcome confirms that the classification signal—though subtle—can be captured by a discriminative model when exposed to labelled examples during training.

The traditional ML pipeline, consisting of a HashingVectorizer and a PassiveAggressiveClassifier, also performed strongly. With an F1 score of 0.972, it rivaled the fine-tuned transformer despite relying solely on sparse feature representations and linear decision boundaries. This result highlights that surface-level textual features may encode sufficient information to distinguish between human and machine authorship in abstracts, possibly due to differences in vocabulary frequency, sentence structure, or lexical density.

5 Discussion

The results of our experiments reveal a notable pattern in the performance of the classification models: large language models (LLMs), including cutting-edge systems such as GPT-4.1 and Qwen3-235B, consistently performed near chance level in distinguishing between human- and machine-written scientific abstracts. In contrast, both the fine-tuned transformer model and the traditional classifier achieved near-perfect performance, with F1 scores of 0.974 and 0.972 respectively.

This sharp discrepancy raises several important questions about the nature of the detection task and the limitations of zero-shot LLM inference. The underwhelming results of the LLMs may stem from the zero-shot setup used in the experiments. Although LLMs have demonstrated broad competence in a range of generative and reasoning tasks, their performance in subtle classification settings—particularly without task-specific training—is often limited. In our case, the classification task relies on capturing fine-grained, often imperceptible linguistic differences between two texts that are topically identical and structurally similar. These nuances may not be readily detectable without additional context or calibration.

Another contributing factor is the in-domain similarity of the texts. Since both human- and machine-generated abstracts summarise the same research paper, they often share terminology, structure, and even phrasing. This results in minimal surface-level variation—precisely the kind of variation that LLMs may overlook in the absence of tailored prompting or fine-tuning.

Furthermore, LLMs are inherently generative, not discriminative. When repurposed for binary classification in a zero-shot setting, they rely heavily on probabilistic reasoning and internal priors, which may not be accurate for a highly specific detection task such as this. Their inability to identify stylistic markers of synthetic writing without explicit examples severely limits their utility in authorship verification.

The success of both the traditional PassiveAggressive classifier and the LoRA-fine-tuned DistilBERT suggests that authorship signals do exist in the data, but they are subtle and best captured by models with explicit supervision. The dataset shows a consistently higher word count in the generated versions by domain, which may have been a clear indicator for these models. There may be some lexical patterns such as "This paper/study/review presents/examines/provides…"

The PassiveAggressive classifier, leveraging a simple bag-of-words approach, likely benefits from capturing statistical regularities in vocabulary use, lexical density, or syntactic patterns that differ—perhaps subtly but consistently—between human and machine writers. These cues might include phrase redundancy, sentence-initial tokens, or unnatural repetition that are hard to detect perceptually but easily exploited by statistical models.

The DistilBERT model, fine-tuned via LoRA, excels likely because it is explicitly trained on the classification objective, allowing it to learn nuanced distinctions over multiple layers of abstraction. The results highlight the value of supervised discriminative learning even in tasks where the classes appear nearly indistinguishable to a human reader or an unadapted LLM.

These findings carry significant implications:

- The detection of AI authorship may not require deep semantic modelling, but rather benefits from the exploitation of shallow stylistic inconsistencies. This opens opportunities for lightweight, interpretable, and resource-efficient detection systems.
- Future detection strategies should consider ensemble approaches, combining the broad generalisation of LLMs with the precision of discriminative classifiers.

6 Conclusions and Future Work

This study investigated the detection of AI-generated content in scientific abstracts by evaluating a range of modelling strategies, including zero-shot prompting of large language models (LLMs), fine-tuned transformer encoders, and traditional machine learning classifiers. Surprisingly, the most advanced LLMs—including GPT-4.1 and Qwen3-235B—performed at near-chance levels in the binary classification task. In contrast, a lightweight encoder-based model fine-tuned with Low-Rank Adaptation (LoRA) and a traditional PassiveAggressive classifier achieved near-perfect classification accuracy.

These findings suggest that while LLMs excel at text generation and general reasoning, they are not well-suited for fine-grained authorship attribution in a zero-shot setting, especially when the candidate texts share substantial semantic overlap. On the other hand, task-specific supervised approaches—both neural and statistical—are capable of capturing subtle linguistic cues that differentiate human- and machine-generated writing.

Several limitations should be noted: (i) LLMs were tested exclusively in zero-shot mode, without prompt tuning, few-shot examples, or in-context learning strategies; (ii) all synthetic abstracts were produced by GPT-4.1, which may limit generalizability; (iii) the study focused exclusively on English-language abstracts.

Building on this limitations, several promising directions can be pursued:

- Explainable detection: Integrating explainability tools (e.g., SHAP, attention visualisation) into detection pipelines could reveal which linguistic features signal machine authorship and support trust in automated tools.
- Multilingual detection: Expanding the dataset and experiments to include other languages would allow evaluation of AI authorship detection across diverse linguistic and cultural contexts.
- Human-in-the-loop verification: Combining automated detection with expert judgment could yield hybrid frameworks that balance efficiency and reliability in academic publishing workflows.
- Comparison with abstracts from scientific papers published after gen-AI open-source tools, with the purpose of inferring whether automatic writing is being used in scientific writing.

Acknowledgements

This research has been partially funded by the University of Alicante, University of Havana, the Spanish Ministry of Science and Innovation, and the Generalitat Valenciana, through the "The limits and future of data-driven approaches: A comparative study of deep learning, knowledge-based and rule-based models and methods in Natural Language Processing" (CIDEXG/2023/12) project.

References

Jawaher Alghamdi, Suhuai Luo, and Yuqing Lin. 2023. A comprehensive survey on machine learning approaches for fake news detection.

Sai Anirudh Athaluri, Sandeep Varma Manthena, V S R Krishna Manoj Kesapragada, Vineel Yarlagadda, and Tirth Dave. 2023. Exploring the boundaries of reality: Investigating the phenomenon of artificial intelligence hallucination in scientific writing through chatgpt references. *Cureus*, 15(4).

Franz E. Babl and Maximilian P. Babl. 2023. Generative artificial intelligence: Can chatgpt write a quality abstract? *Emergency Medicine Australasia*, 35(5):809–811.

Giuseppe Biondi-Zoccai, Anna Cazzaro, Elisa Cobalchin, Diletta D'Auria, Giovanni Ardizzone, Salvatore Giordano, Ulvi Mirzoyev, Petar M. Seferovic, Gani Bajraktari, and Denisa Muraru. 2025. Artificial intelligence tools for scientific writing: The good, the bad and the ugly. *Top Italian Scientists Journal*, 2(1).

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Marc Forment, Juanan Pereira, Francisco García-Peñalvo, Maria Casañ, and Jose Cabré. 2025. Lamb: An open-source software framework to create artificial intelligence assistants deployed and integrated into learning management systems. *Computer Standards Interfaces*, 92:103940.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text.

Ehtesham Hashmi, Sule Yildirim Yayilgan, Muhammad Yamin, Subhan Ali, and Mohamed Abomhara. 2024. Advancing fake news detection: Hybrid deep learning with fasttext and explainable ai. *IEEE Access*, 12:44462 – 44480.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection.

Jingshan Huang and Ming Tan. 2023. The role of chatgpt in scientific communication: writing better scientific review articles. *American Journal of Cancer Research*, 13(4):1148–1154. Epub April 15, Published April 30.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nathan Jenko, Sisith Ariyaratne, Lee M. Jeys, Scott Evans, Krishna P. Iyengar, and Rajesh Botchu. 2024. An evaluation of ai generated literature reviews in musculoskeletal radiology. *The Surgeon*, 22(3):194–197.

Sarah Kreps, Miles McCain, and Miles Brundage. 2022. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, 9:104–117.

Edisa Lozić and Benjamin Štular. 2023. Fluent but not factual: A comparative analysis of chatgpt and other ai chatbots' proficiency and originality in scientific writing for humanities. *Future Internet*, 15(10):336.

Tanjim Mahmud, Imran Hasan, Mohammad Tarek Aziz, Taohidur Rahman, Mohammad Shahadat Hossain, and Karl Andersson. 2024. Enhanced fake news detection through the fusion of deep learning and repeat vector representations. In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pages 654–660.

Eleftheria Papageorgiou, Christos Chronis, Iraklis Varlamis, and Yassine Himeur. 2024. A survey on the use of large language models (llms) in fake news. *Future Internet*, 16(8).

Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2024. Hc3 plus: A semantic-invariant human chatgpt comparison corpus.

A Comparative Study of Hyperbole Detection Methods: From Rule-Based Approaches through Deep Learning Models to Large Language Models

Silvia Gargova[1] **Nevena Grigorova**[1] **Ruslan Mitkov**[2]
[1]Big Data for Smart Society Institute (GATE), Bulgaria,
[2]University of Alicante, Spain
{silvia.gargova, nevena.grigorova}@gate-ai.eu
ruslan.mitkov@ua.es

Abstract

We address hyperbole detection as a binary classification task, comparing rule-based methods, fine-tuned transformers (BERT, RoBERTa), and large language models (LLMs) in zero-shot and few-shot prompting (Gemini, LLaMA). Fine-tuned transformers achieved the best overall performance, with RoBERTa attaining an F1-score of 0.82. Rule-based methods performed lower (F1 = 0.58) but remain effective in constrained linguistic contexts. LLMs showed mixed results: zero-shot performance was variable, while few-shot prompting notably improved outcomes, reaching F1-scores up to 0.79 without task-specific training data. We discuss the trade-offs between interpretability, computational cost, and data requirements across methods. Our results highlight the promise of LLMs in low-resource scenarios and suggest future work on hybrid models and broader figurative language tasks.

1 Introduction

Hyperbole, a common figure of speech that involves deliberate exaggeration, plays an important role in natural language communication by conveying emphasis, emotion, and humor. Detecting hyperbole automatically is a challenging yet valuable task for natural language processing (NLP), with applications in sentiment analysis, position detection, sarcasm recognition, and computational humor. Despite its linguistic and practical significance, hyperbole detection remains underexplored compared to related figurative language phenomena such as metaphor, irony, and sarcasm (Zhang and Wan, 2021; Troiano et al., 2018; Zhang et al., 2024).

The first systematic work on hyperbole detection was carried out by Troiano et al. (2018), who introduced HYPO, the first dataset dedicated to exaggeration detection. Their study framed hyperbole detection as a supervised binary classification problem and demonstrated that semantic features, particularly those that capture quantity and quality, two core linguistic dimensions of exaggeration, enabled traditional classifiers such as logistic regression to achieve beyond chance performance. However, these early rule-based and feature-engineered approaches, although interpretable, suffered from limited generalizability and required extensive linguistic knowledge (Chen et al., 2022; Oprea and Magdy, 2019; Eke et al., 2021).

The field progressed with the adoption of deep learning methods, motivated by the need for richer semantic representations. Early neural models such as CNNs and LSTMs provided moderate improvements Ghosh and Veale (2016); Chen et al. (2022), and Kong et al. (2020) demonstrated that deep learners could substantially outperform traditional models. Their introduction of HYPO-cn, a Chinese dataset, further expanded the scope of research, showing that LSTM-based systems combining handcrafted and embedding features achieved up to 85.4% accuracy.

A major breakthrough came with the advance of transformer-based models. Fine-tuning BERT on the HYPO dataset improved accuracy to 80% (Zhang and Wan, 2021), significantly surpassing earlier methods and confirming the effectiveness of learned contextual representations for hyperbole detection. Further refinements, such as multitask training with literal paraphrases, achieved additional gains (Biddle et al., 2021; Schneidermann et al., 2023).

More recently, research has turned towards large language models (LLMs). While LLMs such as LLaMA, BLOOM, and ChatGPT exhibit strong general-purpose language understanding, studies show that their zero-shot hyperbole detection performance is weak, reflecting an incomplete grasp of this figurative device (Badathala et al., 2023). Even when able to recognise prototypical hyperboles,

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 30–38, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-004

LLMs struggle with cases involving overlap with metaphors or context-dependent exaggeration. To address these shortcomings, recent work explores advanced prompting techniques (Zheng et al., 2025; Xu et al., 2024) and hybrid approaches combining LLMs with human expertise, rule-based verification, or emotion-aware modules (Cohen et al., 2025; Qu et al., 2024).

In this paper, we conduct a comprehensive comparison of three distinct approaches for hyperbole detection: (1) a handcrafted rule-based system, (2) fine-tuned transformer models, and (3) prompt-based inference with LLMs in zero-shot and few-shot settings. Our evaluation on benchmark data reveals the strengths and limitations of each paradigm in terms of accuracy, computational efficiency, and generalisability. We show that while fine-tuned transformers achieve the highest performance, LLMs offer competitive results with minimal task adaptation, and rule-based methods remain viable in constrained scenarios.

The contributions of this study are as follows: (i) an empirical analysis and comparative evaluation of hyperbole detection using diverse methodologies ranging from rule-based methods through deep learning models to large language models, (ii) a thorough evaluation of prompt-based LLMs applied to this task, and (iii) insight into the strengths and limitations of each method within the task's challenging landscape.

The rest of the paper is structured as follows. Section 2 overviews related work. Section 3 details the data used in this study. Section 4 presents the experimental setup, outlining the approaches employed, while Section 5 reports the evaluation results. Section 6 offers discussion of the results. Finally, Section 7 summarises the main findings and proposes future research directions.

2 Related Work

While NLP has long studied figurative language phenomena such as metaphor, irony, and sarcasm, hyperbole detection has only recently emerged as a dedicated research topic. It was largely overlooked until Troiano et al. (2018) introduced the HYPO dataset, the first corpus of hyperbolic and literal sentences. Their work framed hyperbole detection as a supervised binary classification problem and demonstrated that handcrafted features grounded in linguistic theory—particularly quantity and quality distinctions—enabled traditional classifiers such as logistic regression to achieve up to 76% F1 score when literal paraphrases were used as negative examples.

Early approaches mainly relied on rule-based methods and lexical heuristics (Burgers et al., 2016). These methods exploited cues such as extreme adjectives, interjections, or polarity intensification (Kunneman et al., 2015) to identify exaggerations. While interpretable, such systems were brittle and lacked scalability to diverse real-world data. The release of HYPO enabled systematic experimentation with machine learning methods, establishing a foundation for subsequent research.

The next wave of studies adopted neural models, motivated by their ability to capture deeper semantic information. Ghosh and Veale (2016) explored early neural network architectures, while Kong et al. (2020) showed that deep learning approaches substantially outperformed feature-based models. Their work introduced HYPO-cn, a Chinese dataset, and demonstrated that an LSTM-based model could achieve 85.4% accuracy by integrating embeddings with handcrafted features.

Transformer-based models soon set the state of the art. Zhang and Wan (2021) reported that fine-tuning BERT on HYPO improved accuracy to 80%, a significant leap over the best traditional baseline of 72%. Biddle et al. (2021), Badathala et al. (2023) and Schneidermann et al. (2023) extended this line of research by using multitask learning and literal paraphrases as privileged information, showing that transformers could exploit more nuanced contextual signals.

More recently, researchers have evaluated LLMs such as LLaMA, BLOOM, and ChatGPT for hyperbole detection. Although these models perform well on a wide range of NLP tasks, their zero-shot performance on hyperbole classification is poor, revealing a limited understanding of exaggeration (Badathala et al., 2023). Even ChatGPT, which can correctly classify prototypical hyperboles, struggles with multi-class cases involving metaphor-hyperbole overlaps. To improve LLM performance, studies have investigated advanced prompting methods, including chain-of-thought reasoning, which helps models articulate reasoning but still fails to capture the emotional and contextual subtleties of hyperbole (Zheng et al., 2025; Xu et al., 2024).

Split	Label	Source	# sentences	Total per label	Total per split
train	0	HYPO L	1979	2917	5834
		HYPO - literal	469		
		HYPO - paraphrase	469		
		HYPO - hyperbole	469		
	1	HYPO L	767	2917	
		HYPO XL	1681		
dev	0	HYPO L	120	360	720
		HYPO - literal	120		
		HYPO - paraphrase	120		
		HYPO - hyperbole	120		
	1	HYPO L	120	360	
		HYPO XL	120		
test	0	HYPO L	120	360	720
		HYPO - literal	120		
		HYPO - paraphrase	120		
		HYPO - hyperbole	120		
	1	HYPO L	120	360	
		HYPO XL	120		
				Total:	**7274**

Table 1: Data splits.

3 Data

In this section, we describe the datasets used in our experiments, along with the procedure for splitting the data into training, development, and test sets.

3.1 Used datasets

We used three existing datasets: HYPO (Troiano et al., 2018), HYPO L, and HYPO XL [1].(Zhang and Wan, 2021)

HYPO contains 2,127 sentences, with 709 examples of hyperbole and 1,418 without. Of the non-hyperbolic sentences, 709 are literal paraphrases of the hyperbolic ones (where hyperbolic words or phrases were replaced with literal equivalents). The remaining 709 non-hyperbolic sentences feature the same phrases in their literal sense.

HYPO L consists of 3,226 sentences: 1,007 with hyperbole and 2,219 without. These sentences were first automatically annotated and then human-verified for accuracy.

HYPO XL is made up of 17,862 automatically annotated sentences, all of which contain hyperbole.

3.2 Data splits

The original datasets exhibit a high degree of class imbalance. To enable robust training and evaluation, we constructed a balanced dataset through a two-stage process.

In the first stage, we merged the HYPO and HYPO L datasets and recast the task as binary classification, assigning a label of 1 to hyperbolic sentences and 0 to non-hyperbolic ones. We then sampled additional hyperbolic instances from HYPO XL to achieve an equal number of examples for each class in the combined dataset.

In the second stage, we partitioned the data into training, development, and test sets. Both the development and test sets are perfectly balanced, containing 720 sentences each—360 hyperbolic and 360 non-hyperbolic—while also maintaining an equal distribution across the original data sources. The training set consists of the remaining 5,834 sentences, evenly split between the two classes. However, in contrast to the development and test sets, the distribution of examples across data sources in the training set is not uniform.

Table 1 summarises the size and composition of each data split.

4 Experimental setup

We frame hyperbole detection as a binary sentence classification task, where each input sentence is labeled as either hyperbolic or non-hyperbolic. In this section, we describe the model architecture,

[1] https://github.com/yunx-z/MOVER

training configuration, and evaluation methodology used in our experiments. The objective is to compare the performance of a rule-based approach with two deep learning models and two large language models for the task of hyberbole detecton.

4.1 Rule-based method

Our rule-based approach to automatic hyperbole detection integrates lexical, syntactic, and semantic cues derived from established linguistic resources and syntactic analyses. The system leverages a combination of handcrafted lexicons, pattern matching, and semantic incongruity detection to identify exaggerated language indicative of hyperbole.

4.2 Data Preprocessing and Linguistic Analysis

For linguistic analysis, input sentences are processed using Stanza POS tagging (Qi et al., 2020), which provides tokenization, part-of-speech tagging, lemmatisation, and dependency parsing. This comprehensive linguistic annotation enables precise syntactic and semantic analysis necessary for detecting subtle forms of exaggeration.

4.2.1 Lexical Resources and Hyperbole Lexicons

We curate several lexicons capturing common hyperbolic expressions across various semantic domains. To construct these lexicons, we manually reviewed the training set exclusively, deliberately excluding the development and test sets to avoid bias.

- **Quantity and Size Adjectives:** Adjectives such as *endless*, *gigantic*, and *limitless* that represent exaggerated quantities or magnitudes.
- **Intense Emotion Verbs and Adjectives:** Verbs and adjectives conveying heightened emotional states (e.g., *die*, *cry*, *terrified*, *ecstatic*), used to detect emotional overstatements.
- **Temporal Exaggerators:** Nouns and adverbs denoting exaggerated durations (e.g., *eternity*, *forever*, *centuries*).
- **Hyperbolic Idiomatic Expressions:** A predefined set of verb-object pairs known to form hyperbolic idiomatic expressions (e.g., *cry me a river*, *break heart*).

4.2.2 Rule-Based Detection of Hyperbolic Patterns

A collection of syntactic and lexical rules is applied to each processed sentence to identify potential hyperbolic cues:

1. **Exaggerated Quantity or Size:** Detection of adjectives from the quantity and size lexicons or large numeric expressions (e.g., *million*, *billion*).
2. **Unrealistic Comparisons:** Identification of comparative constructions typical of hyperbole, such as similes employing patterns like *as . . . as* or *like a*.
3. **Emotional Overstatement:** Recognition of verbs and adjectives associated with intense emotions, with special handling for frequent colloquial hyperbolic phrases (e.g., *so hungry*).
4. **Temporal Exaggeration:** Detection of temporal terms implying extreme duration.
5. **Superlative Forms:** Identification of superlative adjectives (e.g., *biggest*, *most incredible*).

The complete set of rules and the associated lexicons are provided in the Appendix (see Appendix A for the lexicons and Appendix B for the rule set)[2].

4.2.3 Semantic Incongruity Analysis

To complement surface-level rules, we incorporate semantic checks to detect incongruities frequently present in hyperbolic expressions:

- **WordNet Domain Analysis:** Utilizing the WordNet lexical database, semantic domains for verbs and nouns are extracted to assess semantic compatibility. Abstract subjects paired with concrete predicates may signal hyperbole.
- **Verb-Object Selectional Preferences:** By comparing verb domains against expected noun domains, the system flags semantically incongruous verb-object pairs (e.g., *eat horse*).
- **Idiomatic Hyperbole Pairing:** Known idiomatic hyperbolic pairs are matched directly to capture conventionalised exaggerations.

[2] Available at: https://drive.google.com/file/d/1JWRMGPyb7mWrWj0DrEV-JHUjVWge3C_P/view?usp=sharing

4.3 Fine-tuning Transformer Models

For our experiments with transformer-based models, we selected BERT and RoBERTa. For both models, we adopted the standard architecture provided by the Hugging Face Transformers library.

4.3.1 BERT model

Model Architecture We use a standard transformer-based architecture for binary sentence classification, based on the pretrained `bert-base-cased` model from the Hugging Face Transformers library. This version of BERT consists of 12 transformer layers, each with 12 self-attention heads and a hidden size of 768. The model is implemented using the `BertForSequenceClassification` class, which appends a linear classification layer on top of the [CLS] token representation to predict one of two class labels: *hyperbolic* or *non-hyperbolic*. The model is fine-tuned end-to-end on our task-specific data.

Training Configuration The model is fine-tuned using the AdamW optimiser with a learning rate of 2×10^{-5} and trained for 3 epochs with a batch size of 16. A linear learning rate scheduler without warm-up steps is used throughout training. Sentences are tokenised using the `BertTokenizer`, with all inputs truncated or padded to a maximum length of 128 tokens.

Evaluation Methodology Model performance is assessed on the test set. We report standard classification metrics, including **accuracy**, **precision**, **recall**, and **F1-score**, computed using the `scikit-learn` library. Evaluation is conducted in batches using PyTorch's `no_grad()` context to disable gradient tracking. Predicted labels are stored alongside the gold labels to support detailed error analysis.

4.3.2 RoBERTa model

Model Architecture We use the multilingual `XLM-RoBERTa base` model for our experiments, treating the task as a binary sentence classification problem. The model follows a standard transformer encoder architecture, consisting of 12 layers, each with 768 hidden units and 12 self-attention heads. On top of the transformer backbone, a classification head is added—a fully connected layer followed by a softmax layer that outputs a probability distribution over two classes: hyperbolic and non-hyperbolic.

Training Configuration The model is fine-tuned using the Hugging Face Transformers library. Input texts are tokenised using the corresponding `AutoTokenizer`, with truncation and padding applied to ensure a maximum sequence length of 128 tokens. Training is performed using the AdamW optimiser with a learning rate of 2e-5, over 3 epochs, and with a batch size of 16. A linear learning rate scheduler without warm-up steps is employed. The model is trained using the cross-entropy loss.

Evaluation Methodology We evaluate model performance on both the development and test sets using standard classification metrics: accuracy, precision, recall, and F1-score. Predictions are obtained by selecting the class with the highest softmax probability.

4.4 LLM-based methods

For the large language model (LLM) experiments, we evaluated two instruction-tuned models: **Gemini** (proprietary, accessed via API) and **LLaMA** (open-weights, accessed via the Hugging Face Transformers library). Both models were tested in *zero-shot* and *few-shot* configurations. The task required the model to predict whether a given sentence contains hyperbole, returning either `"hyperbole"` or `"not hyperbole"`.

4.4.1 Prompting Strategies

In the *zero-shot* setting, each model was given only a natural language instruction along with the input sentence, as shown below:

```
You are a helpful assistant for
    ↪ detecting hyperbole.

Classify the following text into one of
    ↪ two categories: hyperbole or not
    ↪ hyperbole.

Hyperbole is a figure of speech that
    ↪ uses extreme exaggeration to
    ↪ emphasize a point or create a
    ↪ strong impression. It is not
    ↪ meant to be taken literally and
    ↪ is often used for humor or
    ↪ dramatic effect.

Output only the predicted label (either
    ↪ hyperbole or not hyperbole) and
    ↪ nothing else.

Now classify the following text:

Text: {text}
Classification:
```

In the *few-shot* setting, the prompt included the same instruction followed by several example text–label pairs, illustrating both hyperbolic and non-hyperbolic cases. An example prompt is given below:

```
You are a helpful assistant for
    ↪ detecting hyperbole.

Classify the following text into one of
    ↪ two categories: hyperbole or not
    ↪ hyperbole.

Hyperbole is a figure of speech that
    ↪ uses extreme exaggeration to
    ↪ emphasize a point or create a
    ↪ strong impression. It is not
    ↪ meant to be taken literally and
    ↪ is often used for humor or
    ↪ dramatic effect.

Here are some examples:

{examples}

Now classify the following text:

Text: {text}
Classification:
```

The examples were selected to cover a range of syntactic and semantic structures typically associated with hyperbolic and literal expressions.

4.4.2 Inference Parameters

To ensure consistent model behaviour across conditions, we fixed the following decoding parameters:

- **Temperature:** 0 (to enforce deterministic output)
- **Max tokens:** 5 (to limit responses to concise labels)

Gemini was accessed via its official API, while LLaMA was executed locally using the Hugging Face Transformers interface with identical prompt structures and generation settings.

The outputs from both models were normalised to binary labels, with `"hyperbole"` mapped to the positive class (1) and `"not hyperbole"` to the negative class (0). Any non-standard outputs were either discarded or resolved using simple pattern matching heuristics.

4.4.3 Evaluation Protocol

All LLM outputs were evaluated against the held-out test set used consistently across all models. We computed standard classification metrics, including accuracy, precision, recall, and F1-score. This allowed for direct and fair comparison with both the rule-based baseline and the fine-tuned transformer models.

5 Results

This section presents the evaluation outcomes of the tested approaches on the classification task. We report performance metrics across different experimental setups. The results provide insights into the effectiveness and comparative strengths of each method.

The results, summarised in Table 2, show a clear performance difference across the evaluated methods.

5.1 Rule-Based method

The rule-based method was built using a set of manually designed rules based on common patterns found in hyperbolic expressions—for example, extreme adjectives, intensifiers, or emotional phrases. This system achieved an accuracy of 56%, a precision of 0.55, a recall of 0.60, and an F1-score of 0.58.

These results indicate that the rule-based system is capable of detecting certain prototypical cases of hyperbole, particularly when the language follows well-defined and recognisable patterns. However, its performance declines when faced with more subtle, context-dependent, or creatively expressed instances. This suggests that while rule-based approaches can offer interpretability and precision in constrained settings, they lack the flexibility needed to generalise across the diverse and often ambiguous forms of hyperbolic language found in natural discourse.

While the overall performance is relatively low compared to machine learning models, the rule-based system is still useful. It provides insight into which linguistic features are most important for hyperbole and serves as a transparent and interpretable baseline.

5.2 Fine-Tuned Transformer Models

Both BERT and RoBERTa performed much better than the rule-based system.

- **BERT** achieved an accuracy of 81%, with precision of 0.86 and an F1-score of 0.80. BERT tends to be cautious, favouring precision over recall. This means it is good at avoiding false

Method	Model	Accuracy	Precision	Recall	F1-score
Rule-based	-	0.56	0.55	0.60	0.58
Fine-Tuned Transformer	BERT	0.81	0.86	0.75	0.80
Fine-Tuned Transformer	**RoBERTa**	**0.82**	0.81	0.83	**0.82**
LLM zero-shot	Gemini-2.5-flash-lite	0.71	0.80	0.58	0.67
	Meta-Llama-3-8B-Instruct	0.68	0.80	0.47	0.59
LLM few-shot	Gemini-2.5-flash-lite	**0.78**	0.80	0.78	**0.79**
	Meta-Llama-3-8B-Instruct	0.74	0.68	0.88	0.77

Table 2: Results on the test set.

positives, which is helpful in situations where incorrect detection could be problematic.

- **RoBERTa** performed slightly better than BERT. It achieved the best overall results, with an accuracy of 82%, recall of 0.83, and F1-score of 0.82. RoBERTa was better at finding true cases of hyperbole (higher recall) while still keeping precision high, possibly due to its stronger pre-training.

5.3 Large Language Models (LLMs)

We also tested large instruction-tuned models, Gemini and LLaMA, in zero-shot and few-shot settings. These models were not fine-tuned on our dataset, but we gave them task instructions (and a few examples, in the few-shot setting) at inference time.

- **Gemini Zero-Shot** had moderate performance, with accuracy of 72% and F1-score of 0.67. It was highly precise (0.80) but missed many true cases (recall: 0.58), meaning it was conservative in predicting hyperbole.
- **Gemini Few-Shot** improved significantly. With just a few examples, its accuracy rose to 80%, and its F1-score reached 0.79, showing that few-shot prompting can help LLMs better understand the task.
- **LLaMA Zero-Shot** had weaker performance, with a low recall of 0.46 and F1-score of 0.60, even though precision remained high (0.80). Like Gemini, it was overly cautious.
- **LLaMA Few-Shot** improved the most in recall (0.88), meaning it detected many true hyperboles, but at the cost of lower precision (0.68) and more false positives. This suggests it became overconfident in labelling hyperbole after seeing a few examples.

6 Discussion

Among all models, **RoBERTa** achieved the highest overall performance (F1 in the low 80s), highlighting the effectiveness of fine-tuned transformer models for hyperbole detection. **BERT** and **Gemini Few-Shot** also performed competitively (F1 in the high 70s), showing that both supervised learning and few-shot prompting can yield strong results. Although the gap between fine-tuned transformers and few-shot LLMs is relatively small, it is practically meaningful: supervised transformers consistently generalise better across data splits, while few-shot LLMs offer flexible, annotation-free alternatives that trade a few points of accuracy for drastically lower requirements in labelled data.

While the rule-based method performed less well in aggregate metrics (F1 around the high 50s), it remains valuable in certain settings. One of the main challenges we faced was the difficulty of capturing the full range of hyperbolic expressions through a fixed set of handcrafted rules. Hyperbole often relies on creative, context-dependent language, which makes it hard to exhaustively define through linguistic patterns alone. As a result, the system struggled with generalisation and coverage. Nevertheless, in highly constrained domains where hyperbolic forms are stable and predictable, such systems may perform comparably to neural approaches, especially when transparency and efficiency are prioritised.

Although the zero-shot LLMs (Gemini and LLaMA) were less accurate overall (F1 in the high 50s), they show strong potential in low-resource settings. Their performance improves significantly with just a few examples, making them flexible tools for tasks where annotated data is limited or unavailable. Nevertheless, LLMs are computationally expensive to run and may produce inconsistent outputs depending on prompt design and input formulation.

Taken together, these results highlight a trade-off between performance, data requirements, and computational costs: fine-tuned transformers deliver the strongest accuracy but require labelled data; few-shot LLMs offer near-competitive results with minimal annotation; and rule-based systems, though weakest in absolute performance, provide efficiency and interpretability in specialised contexts.

7 Conclusion and Future work

This paper casts the task of hyperbole detection as a binary classification problem, comparing rule-based methods, fine-tuned transformer models, and large language models (LLMs) in both zero-shot and few-shot configurations. Our findings demonstrate that fine-tuned transformer models—particularly RoBERTa—offer the most robust performance overall, with F1 scores in the low 80s, clearly outperforming both handcrafted rule systems and prompt-based LLMs across standard evaluation metrics.

The relative performance differences are significant: while few-shot LLMs achieved F1 in the high 70s, suggesting they are competitive with fine-tuned transformers in practical terms, their advantage lies in requiring no annotated training data. By contrast, zero-shot LLMs and rule-based methods, both yielding F1 in the high 50s, lag behind in predictive accuracy but retain value in specific conditions—such as absence of labelled data, domain-specific constraints, or the need for interpretability. This performance spectrum indicates that model choice should be guided by resource availability and task requirements rather than accuracy alone.

Future work could explore hybrid approaches that combine the interpretability of rule-based systems with the generalisability of neural models. In addition, improving prompt engineering strategies and model calibration may further enhance the reliability of LLMs in zero-shot settings. Finally, expanding the task to include more nuanced figurative language phenomena, such as irony, may offer a more comprehensive understanding of exaggeration in natural language.

Acknowledgments

This work is supported by GATE project funded by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 programme under grant agreement no. 857155, the programme "Research, Innovation and Digitalization for Smart Transformation" 2021-2027 (PRIDST) under grant agreement no. BG16RFPR002-1.014-0010-C01, and the BROD project, funded by the Digital Europe programme of the European Union under grant agreement no. 101083730.

This work has also been partially supported by the CIDEXG/2023/12 project, funded by the Generalitat Valenciana.

References

Naveen Badathala, Abisek Rajakumar Kalarani, Tejpalsingh Siledar, and P. Bhattacharyya. 2023. A match made in heaven: A multi-task framework for hyperbole and metaphor detection. *Annual Meeting of the Association for Computational Linguistics*.

Rhys Biddle, Maciek Rybinski, Qian Li, Cecile Paris, and Guandong Xu. 2021. Harnessing privileged information for hyperbole detection. In *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pages 58–67, Online. Australasian Language Technology Association.

Christian Burgers, Britta C. Brugman, Kiki Y. Renardel de Lavalette, and Gerard J. Steen. 2016. Hip: A method for linguistic hyperbole identification in discourse. *Metaphor and Symbol*, 31(3):163–178.

WangQun Chen, Fuqiang Lin, Xuan Zhang, Guowei Li, and Bo Liu. 2022. Jointly learning sentimental clues and context incongruity for sarcasm detection. *IEEE Access*.

Kevin Cohen, Laura Manrique-G'omez, and Rub'en Manrique. 2025. Historical ink: Exploring large language models for irony detection in 19th-century spanish. *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*.

C. Eke, A. Norman, and Liyana Shuib. 2021. Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and bert model. *IEEE Access*.

Aniruddha Ghosh and T. Veale. 2016. Fracking sarcasm using neural network. *WASSA@NAACL-HLT*.

Li Kong, Chuanyi Li, Jidong Ge, B. Luo, and Vincent Ng. 2020. An empirical study of hyperbole. *Conference on Empirical Methods in Natural Language Processing*.

Florian Kunneman, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management*, 51(4):500–509.

Silviu Vlad Oprea and Walid Magdy. 2019. isarcasm: A dataset of intended sarcasm. *Annual Meeting of the Association for Computational Linguistics*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Xingwei Qu, Ge Zhang, Siwei Wu, Yizhi Li, and Chenghua Lin. 2024. Overview of the nlpcc 2024 shared task on chinese metaphor generation. *Natural Language Processing and Chinese Computing*.

Nina Schneidermann, Daniel Hershcovich, and Bolette S. Pedersen. 2023. Probing for hyperbole in pre-trained language models. *Annual Meeting of the Association for Computational Linguistics*.

Enrica Troiano, C. Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. *Conference on Empirical Methods in Natural Language Processing*.

Yanzhi Xu, Yueying Hua, Shichen Li, and Zhongqing Wang. 2024. Exploring chain-of-thought for multimodal metaphor detection. *Annual Meeting of the Association for Computational Linguistics*.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. Sarcasmbench: Towards evaluating large language models on sarcasm understanding. *arXiv.org*.

Yunxiang Zhang and Xiaojun Wan. 2021. Mover: Mask, over-generate and rank for hyperbole generation. *North American Chapter of the Association for Computational Linguistics*.

Limin Zheng, Sihang Wang, Hao Fei, Zuquan Peng, Fei Li, Jianming Fu, Chong Teng, and Donghong Ji. 2025. Enhancing hyperbole and metaphor detection with their bidirectional dynamic interaction and emotion knowledge. *Annual Meeting of the Association for Computational Linguistics*.

Evaluating the Performance of Transformers in Translating Low-Resource Languages through Akkadian

Daniel Jones
Lancaster University, United Kingdom
d.jones15@lancaster.ac.uk

Ruslan Mitkov
University of Alicante, Spain
ruslan.mitkov@ua.es

Abstract

In this paper, we evaluate the performance of various fine-tuned, transformer-based models in translating Akkadian into English. Using annotated Akkadian data, we seek to establish potential considerations when developing models for other low-resource languages, which do not yet have as robust data. The results of this study show the potency, but also cost ineffeciency of Large Language Models compared to smaller Neural Machine Translation models. Significant evidence was also found demonstrating the importance of fine-tuning machine translation models from related languages.

Keywords: transformer, neural machine translation, low-resource, Akkadian

1 Rationale

Ancient languages serve as vital links to our cultural and historical heritage. Akkadian, once the lingua franca of Mesopotamia (George, 2007), exemplifies this connection. Massive digitisation initiatives such as ORACC and the CDLI projects have generated extensive corpora of transliterated cuneiform texts (Tinney et al., 2025; CDLI Contributors, 2025); for instance, the Ur III corpus comprises over 72,000 transcribed texts, yet only 2.2% have been translated into modern languages (Punia et al., 2020). This stark bottleneck underscores the need for robust machine translation (MT) tools that can democratise access to these historical records for Assyriologists and scholars alike.

Since the introduction of the Transformer architecture (Vaswani et al., 2017), MT solutions have swiftly moved away from statistical models in favour of neural approaches. The surge in academic interest in low-resource languages – characterised by their limited digital presence and sparse representation in training data – has further highlighted the challenges faced by contemporary Large Language Models such as ChatGPT or Gemini in low-resource scenarios (Hasan et al., 2024). These languages necessitate tailored strategies to achieve robust translations, whether through fine-tuning on curated sentence pairs or via methods like Retrieval Augmented Generation (RAG), as explored by Shu et al. (2024). However, this study focuses exclusively on fine-tuning.

Akkadian itself, though extinct and largely confined to the realm of Assyriology, benefits from a uniquely rich, highly annotated corpus[1] with many bidirectional translations – an advantage seldom seen in low-resource languages. This abundance of quality data obviates the need for extensive data augmentation techniques often employed in projects for under-documented languages (e.g., as described in NLLB, 2022). Instead, Akkadian offers an ideal testbed for evaluating the performance of different pre-trained transformer architectures – both sequence-to-sequence (seq2seq) models and causal (decoder-only) models – in a low-resource, morphologically distinct setting. Additionally, the relative simplicity of the cuneiform transliteration system, in which wedge clusters represent syllables (Schmandt-Besserat, 2014), enables a straightforward conversion into the Latin alphabet, making Akkadian particularly amenable to phonetic-like translation tasks.

Our investigation evaluates how model architecture, parameter count, and the nature of pretraining data (including exposure to related Semitic languages) influence translation quality. By establishing a performance baseline for Akkadian-to-English translation, our study not only addresses a critical gap in the digitisation and translation of ancient texts but also lays the groundwork for broader applications of MT for low-resource languages.

[1] Entries in Oracc contain descriptions such as line rulings, which are not necessary regarding this paper, but may be useful for future research into Akkadian OCR

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 39–47, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-005

The remainder of this paper is structured as follows. Section 2 provides a background on related work, including an overview of Akkadian, the digitisation efforts of cuneiform texts, and research on MT for low-resource languages. Section 3 details the methodologies used to fine-tune and train the models, along with the compromises made during data preparation, and the metrics used for evaluation. Section 4 offers a comprehensive evaluation of the results, and Section 5 and 6 conclude with a discussion of the study's implications and potential directions for future research.

2 Preliminaries

2.1 Akkadian

Akkadian is an extinct East Semitic language that was spoken in Mesopotamia – roughly corresponding to modern-day Iraq. Historically, it was written in cuneiform on clay tablets using a wedge-based script. Each cuneiform symbol represents a syllable; for example, to write the word "cat," the script would use two distinct symbols, representing "ca" and "at" respectively (as illustrated by examples from the British Museum). The transliteration of these cuneiform texts into the Latin script forms the source for our machine translation (MT) task.

Akkadian qualifies as a low-resource language due to its limited online presence (Magueresse et al., 2020). For the majority of low-resource languages, this results in difficulty obtaining good training data – ideally a dataset of parallel sentences. Recent efforts to leverage technology for preserving cultural heritage and enhancing digital inclusivity (Galla, 2018; Joshi et al., 2020) have elevated interest in such languages. Notably, collaborative projects like CDLI and Oracc have been instrumental in digitising vast collections of clay tablets, thereby providing a rich bilingual corpus that is rarely available for other low-resource languages.

There are a few important caveats when using Akkadian as a benchmark. Although Akkadian was originally written in cuneiform, its digitised representation in CDLI and Oracc is transliterated into the Latin script. Consequently, the performance of our MT models specifically reflects translation challenges for low-resource languages presented in this format. Additionally, Akkadian occasionally incorporates Sumerian elements – sumerograms or logograms – into its script. For instance, the Akkadian word for "king" is pronounced "sharum" yet may be rendered with the Sumerian term "Lugal." These instances are statistically infrequent and unlikely to significantly affect overall model performance.

This study leverages Akkadian's unique position as a well-annotated yet low-resource language to evaluate and refine neural machine translation techniques, ultimately contributing to both the preservation of ancient cultural heritage and the advancement of MT for underrepresented languages.

2.2 Low-Resource Languages

The digital preservation of ancient languages is vital not only for maintaining the cultural heritage of communities but also for tapping into a vast potential market – after all, there are nearly 3 billion speakers of low-resource languages worldwide (Kshetri, 2024). Initiatives such as Meta's No Language Left Behind (NLLB Team, 2022) have shown that investment in multilingual translation goes beyond charity; it opens up entire emerging markets while preserving unique cultural identities. In this context, Akkadian stands out as a particularly interesting case study. Its status as a low-resource language is compounded by the fact that our source material is transliterated text – the conversion of ancient cuneiform (originally inscribed on clay tablets using wedge impressions) into the Latin alphabet. An example of this transliteration process is evident in the texts provided by Oracc (2025).

2.3 Low-Resource Comparisons and Cuneiform Translation

Solutions to translate Akkadian have been explored previously. Krueger (2023a) details the development of an AI Cuneiform Corpus – a resource for Assyriology that leverages a fine-tuned T5 transformer model to generate translations of both Sumerian and Akkadian texts. His work employs bidirectional translation training, whereby the model is also trained to translate back from English to the source language. This strategy helps stabilise convergence across epochs, even though the work does not report modern metrics, such as BLEU scores, to benchmark its performance. The model's availability on HuggingFace (Krueger, 2023b) enables direct, side-by-side comparisons with other approaches, making it a valuable reference point for our own T5 experiments.

During a period of heightened interest in the machine translation of ancient languages, Punia et al. (2020) evaluated multiple architectures for translating Sumerian to English. Their study compared a Base Translator – an LSTM-based model without pre-trained embeddings – against an Extended Translator that incorporated pre-trained embeddings from the Wikipedia corpus (Pennington et al., 2014), as well as a Transformer-based model. Despite Transformers' well-known advantage in handling long sequences through self-attention, the brevity of cuneiform inscriptions (with an average of just 2.8 tokens per phrase in Punia's dataset) appears to limit the benefits of this architectural choice. As a result, the Extended Translator achieved a slightly higher BLEU score (21.6) compared to the Transformer (20.9), though the Transformer still outperformed the Base Translator. These findings underscore that while self-attention offers robust performance overall, fine-tuning specifics – such as access to pre-trained embeddings – can be particularly crucial in scenarios where input sequences are very short.

Shu et al. (2024) further contribute to this discussion by demonstrating how Retrieval Augmented Generation (RAG) can be used effectively in low-resource settings – in their case, for Cherokee translation. Their RAG model, although yielding moderate BLEU scores, showed impressive semantic understanding as measured by BERTScore. This suggests that even when lexical overlap is low, models can capture deep semantic meaning if properly contextualised through additional data retrieval. While these findings point to the potential of large language models when fine-tuned or augmented appropriately, the higher computational costs involved also highlight the appeal of achieving strong performance through more focused, fine-tuning methods – exactly the approach taken in this project.

3 Data and methodology

3.1 Corpora

A major reason for the choice of Akkadian as the language of interest is the organisation of data that exists. Assyriologists have worked to digitise the world's discovered cuneiform tablets into organised corpora. Since this digitising process occurred over many years, there are inconsistencies within the standardisation of cuneiform, a problem discussed by Krueger (2023a). Some symbols are translated in ASCII, whereas more modern forms maybe transliterated using accented Unicode characters. For this project, data was gathered from the Oracc (2025) and CDLI (2025) corpora. These corpora are extensive and hold enough translation examples to train a competent translator.

3.1.1 Oracc Corpus

The Oracc corpus grew out of recognition of the limitation present in the Electronic Text Corpus of Sumerian Literature (Black et al., 2002). ETCSL was initiated by Jeremy Black and Graham Cunningham of the University of Oxford, and had the ambitious goal to create an online corpus with Sumerian literary texts, along with their English translations (Ebeling, 2007). Though a valuable resource, it was limited in many aspects. Its focus on Sumerian was problematic when considering cuneiform, a writing system used to write countless, linguistically unrelated languages. Furthermore, ETCSL was largely static, limiting the ability of the community to contribute and stunting its development. Recognising these limitations, the Oracc corpus was developed. It allowed for richer annotation, beyond just translations (Oracc, 2019), and emphasised openness and collaboration. Oracc includes glossaries for each subproject within the overall corpus. These glossaries provide information about all the words used within the subproject. While this is not useful for this project, since the model should be able to learn words from exposure during fine-tuning, it does have potential to be useful when considering a technique like RAG (Shu et al., 2024) discussed earlier. These glossaries could be used to scrape a wordlist, which can be used as context for larger models with very potent few-shot capabilities.

3.1.2 CDLI

The Cuneiform Digital Library Initiative (CDLI), unlike Oracc, focuses on being a digital archive for the objects themselves. As such, the tablets have high quality photos and line art, occasionally with transliterations. It has an emphasis on unpublished materials, allowing researchers to access tablets worldwide for research and study. Both CDLI and Oracc provide a valuable resource for this project. Scraping their data allows for transliterated Akkadian to be gathered en masse. Furthermore, it allows for untranslated, but still transliterated, texts to be gathered. While not vital to this project, it is important to a project such as AICC, which used Krueger's model to translate previously undeciphered texts into English.

3.2 Methodology

3.2.1 Models used

The transformers chosen for comparison primarily differ in their architecture, pre-training scope and parameter size. Broadly, the models chosen can be split into Sequence-to-sequence (encoder-decoder) and causal (decoder-only) architectures. The *seq2seq* models used here are considered Neural Machine Translation (NMT) solutions, whereas the *causal* models are considered LLM models. The models chosen were as follows:

- **T5-base** (Raffel et al., 2020): 250 million parameter encoder-decoder model that was used by Krueger (2023a) to translate Akkadian and Sumerian. Uses C4 corpus with mostly English text scraped from the web, and is trained to perform a variety of tasks, including translation. Since this model was used by Krueger, it serves as a baseline and sense check for the quality of our own experiments.

- **MarianMT** (Junczys-Dowmunt et al., 2018): MarianMT has roughly 75 million parameters, and is an encoder-decoder model specifically designed for translation tasks. It has half the layers of T5-base, and is trained on the OPUS corpus (Tiedemann, 2012), which contains parallel corpora in multiple languages. This model provides insight in the tradeoff between fine-tuning a multi-purpose model as opposed to a model designed specifically for translation.

- **Qwen 0.5B-Instruct** (Yang et al., 2024): A 500 million parameter decoder-only model. It is inherently multilingual, being trained across multiple languages. It also utilises some advancements to the transformer architecture, such as Rotary Positional Embeddings, which may allow it to better understand semantics within sentences. It also uses Grouped-Query Attention, which may allow for faster inference times.

- **Mistral 7B** (Lachaux et al., 2023): A 7 billion parameter decoder-only model developed by Mistral AI that leverages Grouped-Query Attention for faster inference. With a parameter count 14 times that of Qwen 0.5B, it offers enhanced few-shot translation performance. The model employs a Byte-Pair Encoding (BPE) tokeniser adapted for translation, ensuring robust handling of both ASCII and Unicode inputs. Training on consumer hardware is made feasible by using quantisation techniques (Gholami et al., 2022) and LoRA (Hu et al., 2021), with most parameters remaining unchanged during fine-tuning.

We seek to establish the performances of these various transformer-based NMT and LLM models – each with different architectures, parameter counts and specifity. As stated, the models chosen for this experiment are MarianMT, T5, Qwen 0.5B instruct, and Mistral 7B. These models are advanced enough to provide worthwhile translations, but can still be trained on consumer hardware, and run cheaply. T5 and MarianMT are encoder-decoder transformers, whereas Qwen and Mistral are decoder-only transformers. The decoder-only architecture not only reduces model size, but also does not need labelled input that a encoder-decoder model might (Fu et al., 2023). This means it can be more readily trained with available data on the internet, hence why it is favoured by the larger language models. This might be of benefit in few-shot capabilities when translating Akkadian to English. Architecturally, Qwen and Mistral use more modern techniques when embedding and using multiple attention heads. Qwen, for example, uses Rotary Positional Embeddings (RoPE). This gives the model an advantage in understanding relative word relations by reducing the influence words have on one another with distance, as opposed to T5's relative positional embeddings, which cannot decay dependencies as effectively (Su et al., 2024). It also benefits faster inference, because of improvements such as Grouped-Query Attention, and has a significantly higher context window then MarianMT and T5.

An important compromise was made in order to train Mistal 7B. Given the limited VRAM on consumer hardware, methods were used to lessen this burden. Namely, through the Unsloth library, Low-Rank Adaptation (LoRA) was used to freeze the original model weights, and only train newly injected matrices. In its introductory paper by Hu et al. (2021), it was shown that LoRA drastically lowers the resources required to train. GPT 175B's VRAM consumption during training using LoRA reduced VRAM consumption form 1.2TB to 350GB. Despite this, empirical evidence has also shown that the fine-tuned capability of LoRA trained models equals, or occasionally outperforms fully-trained models (Agiza et al., 2024; Peters et al., 2019). As such, the outcome can still be considered alongside the other models.

3.2.2 Evaluation Metrics

Translation quality will be assessed using the following metrics:

- **BLEU (Papineni et al., 2002)**: A widely used metric for evaluating machine translation quality, BLEU measures the overlap between n-grams in the generated translation and reference translations, and includes a brevity penalty. It is particularly effective for assessing lexical similarity.
- **BERTScore** (Zhang et al., 2019): This metric evaluates semantic similarity by comparing contextual embeddings of words in the generated translation against those in reference translations. BERTScore measures the accuracy of semantic meaning, making it useful for assessing whether the model captures intended meanings of text, even if the exact words differ.
- **ROUGE2** (Lin, 2004): This metric measures bigram overlap between the generated and reference translations. By capturing both precision and recall of contiguous word sequences, it assesses whether key multi-word expressions and local phrasal structures are preserved. In doing so, ROUGE2 complements BLEU by providing insight into the preservation of phrase-level content.

3.2.3 Gathering

Oracc provides API's to access lists of projects hosted, given in JSON format. Each project has its own collected corpus of cuneiform transliterations that can be navigated through. Each project provides different types of documents, and it is important to appreciate the difference between these when considering a translator. For example, a project such as *akklove* (2025), contains Akkadian love literature. Within this corpus are lengthy poems that incorporate descriptive vocabulary. On the other hand, a corpus such as The Royal Inscriptions of Assyria online (Grayson et al., 2025) contains royal inscriptions, which are often shorter and much more repetitive than *akklove*. It is important to appreciate the bias and diversity of the dataset in order to make best use of it. Data scraped from these projects are in the form of ATF files, a format specifically designed to digitise cuneiform tablets, whilst maintaining metadata about its format.

Krueger (2023b) had already scraped the corpora, and using the same training data provides a foundational benchmark to assess our models against existing solutions. Overall, 95,629 samples were used, with a split of 90% training (86,066) and 10% testing (9,563). A validation set was not used in this case, since tuning of hyperparameters was not performed. Instead, a learning rate of 2e-5 was used for all models, with varying numbers of epochs. This is a common learning rate for fine-tuning transformer models, and was used by Krueger (2023a).

4 Experiments and Results

The models were each trained with a different number of epochs. MarianMT models, along with T5, were trained for 15 epochs, while Qwen 0.5B was trained for 3. Mistral7B was only trained for 1 epoch. This is mostly due to limitations in compute, but provides insight into the few-shot capabilities of the larger models.

4.1 All Sentences

Table 1 reports BLEU, ROUGE-2 and BERTScore over all test sentences.

- Prec - ROUGE2 Precision
- Recall - ROUGE2 Recall
- F1 - ROUGE2 F1

It shows 6 fine-tuned models. The LLM's, Mistral 7B and Qwen 0.5B, as opposed to the NMT models, MarianAr (Arabic→English), MarianEs (Spanish→English), Krueger (T5) and T5. The models are ordered by BLEU score, with Mistral 7B achieving the highest BLEU of 0.478, followed by MarianAr at 0.453, Krueger at 0.416, Qwen at 0.403, T5 at 0.376 and MarianEs at 0.122.

Table 1: Evaluation on all sentences

Model	BLEU	Prec	Recall	F1	BERT
Mistral	0.478	0.527	0.494	0.501	0.930
MarianAr	0.453	0.541	0.508	0.512	0.931
Krueger	0.416	0.530	0.484	0.493	0.930
Qwen	0.403	0.516	0.487	0.491	0.929
T5	0.376	0.420	0.397	0.399	0.914
MarianEs	0.122	0.198	0.303	0.209	0.842

Mistral's 7 billion-parameter model achieved the highest BLEU of 0.478 (3 sf), indicating strong n-gram overlap with the reference. MarianAr followed at 0.453 – a 5.2% deficit – while Krueger's and Qwen trailed at 0.416 and 0.403 respectively. Our T5, trained for only 15 epochs, reached 0.376, and MarianEs (Spanish→English) lagged at 0.122, a 74.5% drop relative to Mistral.

In ROUGE-2 Precision, MarianAr led with 0.541 (54.1% of generated bigrams in the reference), followed by Krueger (0.530), Mistral (0.527) and Qwen (0.516), all within a 4.6% band. T5 and MarianEs fell to 0.420 and 0.198. For Recall, MarianAr attained 0.508, Mistral 0.494, Qwen 0.487 and Krueger 0.484. Combined F1 placed MarianAr at 0.512, Mistral at 0.501 and the next best systems within 4.1%.

On BERTScore most models clustered around 0.930-0.931, effectively within margin of error. MarianAr scored 0.931, with Mistral, Krueger and Qwen at 0.930. T5 scored 0.914 and MarianEs 0.842.

4.2 Long Sentences (Reference $\geq$ 4 words)

Table 2 shows metrics when restricting to sentences with four or more reference tokens.

Table 2: Evaluation on long sentences

Model	BLEU	Prec	Recall	F1	BERT
Mistral	0.473	0.549	0.510	0.519	0.928
MarianAr	0.446	0.562	0.522	0.529	0.929
Krueger	0.407	0.554	0.499	0.510	0.928
Qwen	0.395	0.532	0.496	0.503	0.927
T5	0.370	0.434	0.407	0.409	0.911
MarianEs	0.149	0.230	0.340	0.242	0.854

BLEU dipped slightly for all models (e.g. Mistral from 0.478→0.473, MarianAr 0.453→0.446), while MarianEs rose from 0.122→0.149. Precision increased across the board – MarianAr reached 0.562, Krueger 0.554 – preserving the rank order. Recall and F1 mirrored this improvement (MarianAr recall 0.522, F1 0.529). BERTScore remained effectively unchanged.

4.3 Short Sentences (Reference $<$ 4 words)

Table 3 isolates sentences shorter than four words.

Table 3: Evaluation on short sentences

Model	BLEU	Prec	Recall	F1	BERT
Mistral	0.602	0.408	0.407	0.404	0.938
Qwen	0.593	0.430	0.435	0.428	0.942
Krueger	0.586	0.404	0.406	0.401	0.940
MarianAr	0.541	0.426	0.429	0.423	0.941
T5	0.520	0.346	0.348	0.344	0.927
MarianEs	0.016	0.025	0.103	0.031	0.778

Short sentences boosted BLEU markedly for all except MarianEs: Mistral rose to 0.602, Qwen to 0.593, Krueger to 0.586, each surpassing MarianAr's 0.541. Qwen led ROUGE-2 precision (0.430), recall (0.435) and F1 (0.428). BERTScore peaked at 0.942 for Qwen, with MarianAr at 0.941 and Krueger 0.940.

4.4 Significance Testing

To determine the statistical significance of the differences in BLEU scores between models, we conducted significance tests. Table 4 presents the p-values and confidence intervals for BLEU delta between pairs of models.

Table 4: Significance Test Results (BLEU Delta)

Model 1	Model 2	BLEU Δ Lower CI	BLEU Δ Upper CI	p-value
Krueger	**MarianAr**	–4.10	–3.31	0.00
Krueger	**Mistral7b**	–7.00	–5.68	0.00
Krueger	**MarianEs**	27.72	30.08	0.00
Krueger	**Qwen05**	0.87	1.63	0.00
Krueger	**T5**	3.58	4.32	0.00
MarianEs	**MarianAr**	–33.65	–31.50	0.00
MarianEs	**Mistral7b**	–36.05	–34.31	0.00
MarianEs	**Qwen05**	–28.77	–26.45	0.00
MarianEs	**T5**	–25.98	–23.91	0.00
MarianAr	**Mistral7b**	–3.15	–2.14	0.00
MarianAr	**Qwen05**	4.49	5.39	0.00
MarianAr	**T5**	7.24	8.06	0.00
Mistral7b	**Qwen05**	6.94	8.23	0.00
Mistral7b	**T5**	9.75	10.86	0.00
Qwen05	**T5**	2.28	3.13	0.00

The significance testing results in Table 4 provide a detailed statistical analysis of the differences in BLEU scores between pairs of models. The table includes the lower and upper confidence intervals for the BLEU delta, along with the corresponding p-values. All p-values are less than 0.05, indicating that the differences in BLEU scores between the models are statistically significant.

4.5 Inference Timings

Table 5 compares wall-clock times to translate 500 sentences[2].

Table 5: Inference time for 500 sentences

Model	Total Time (s)	Per Sentence (s)
MarianAr	108	0.22
T5	242	0.48
Qwen	348	0.70
Mistral	1786	3.57

MarianAr was fastest at 108s (0.22s/sentence), over four times quicker than Mistral's 1786s (3.57 s/sentence). T5 and Qwen required 242s and 348s, respectively. Inference speed is an important metric when considering practical applications, and the tradeoff between speed and quality needs to be considered.

4.6 Summary

Overall, MarianAr and Mistral are the top performers: MarianAr leads in all metrics except BLEU (where Mistral narrowly wins). Krueger's T5 surpasses our 15-epoch T5 and Qwen in semantic scores, highlighting the impact of extended training and transfer. Short sentences favour few-shot LLM generalisation (Mistral, Qwen), while longer contexts modestly reduce recall. Finally, smaller specialised models (MarianMT) offer the best tradeoff of speed and quality for practical low-resource language translation.

Shown in Figure 1 are translations between transliterated Akkadian and English, using the MarianAr model.

5 Discussion

When interpreting these results, it's important to remember that our cuneiform corpus is highly repetitive – many near-identical phrases appear in both training and test splits, inflating absolute scores. While this doesn't undermine comparisons between models, it does caution against assuming similar performance on a more varied low-resource dataset.

Firstly, our T5 baseline (15 epochs) underperforms Krueger's T5 (30 epochs) across both BLEU and BERTScore, despite identical hyperparameters. Krueger's additional epochs – and his bidirectional training (English→Akkadian/Sumerian) – helped convergence. This technique is not viable for MarianMT's single-direction architecture, but it signals that smaller models without LLM-style few-shot strength benefit substantially from extended fine-tuning.

Overall, MarianAr (Arabic→English) delivers the best balanced performance. It leads in ROUGE2 and BERTScore, and only narrowly trails Mistral-7B on BLEU. By contrast, MarianEs (Spanish→English) lags dramatically, confirming that even a distantly related Semitic language imbues the model with useful implicit grammatical and lexical knowledge – despite script mismatches and millennia of divergence.

The few-shot prowess of large causal LLMs also shines through. Mistral-7B achieves the top BLEU after a single epoch, and Qwen-0.5B, with three epochs, matches or bests others on very short sentences (<4 words). These results suggest their vast pre-training mitigates sparse data, particularly for lexical matching.

Inference speed highlights practical trade-offs. Although Mistral-7B excels in raw BLEU, its 7 billion parameters slow throughput severely. In contrast, MarianMT variants – especially MarianAr – combine strong quality with sub-second latency, making them better suited for real-world, resource-constrained deployment.

[2] MarianAr was used for timing MarianMT, it can be assumed MarianAr will be roughly equivalent.

Example 1 - Astronomical Text:

Source:
(_mul2_-e)-sza2-_sag_-_gir2-tab 20 si 6 zi ir kur gin ge6 7 sag ge6_ sin ina _igi mul2#_-_kur_-sza2-_kir4_-szil-_pa 3 kusz3_ beta Scorpii
Translation:
The 6th, ZI IR, the east wind blew. Night of the 7th, beginning of the night, the moon was 3 cubits in front of theta Ophiuchi
Example 2 - Royal Inscription:

Source:
(d#)na3#-ku-du-ur2-[ri-uri3] _lugal_ ba-bila2#[(ki)] za-ni-in e2-sag-il2# u3 e2-zi-da# _ibila_ a-sza-re-du sza (d)na3-ibila-uri3 _lugal_ babila2(ki)
Translation:
Nebuchadnezzar, king of Babylon, who provides for the E-sagil and the Ezida, foremost son of Nabopolassar, kingship of Babylonia

Figure 1: Example translations from the MarianAr model showing transliterated Akkadian to English

6 Conclusions and Future Work

In this paper we have demonstrated that careful adaptation of existing NMT architectures can unlock high-quality Akkadian→English translation even under severe data scarcity. Our experiments show that fine-tuning a MarianMT model pre-trained on Arabic (MarianAr) delivers the best balance of surface accuracy (BLEU), semantic fidelity (ROUGE2, BERTScore), and inference efficiency. Despite the millennia that separate Arabic and Akkadian – and the mismatch between Arabic script and romanised transliteration – MarianAr's internalised Semitic grammar and vocabulary proved remarkably transferrable. At the same time, we observed that large causal LLMs such as Mistral-7B and Qwen-0.5B require only one to three epochs to rival or exceed other models on shorter sentences, underlining their potent few-shot adaptation. Yet the hefty parameter counts of these LLMs incur a tangible latency penalty, reaffirming the practical importance of lightweight, specialised NMT when deployment speed and resource budgets are at a premium.

Looking ahead, several avenues promise to extend and deepen these findings. First, pushing MarianAr and our T5 baseline through additional epochs and systematic hyperparameter sweeps will clarify the point of diminishing returns and guard against overfitting. Second, a mixture-of-experts framework – where a fast NMT core handles routine or formulaic passages while a heavyweight LLM tackles longer or more ambiguous sentences – could marry speed with versatility. Third, augmenting our pipeline to ingest raw cuneiform images and output English translations would bridge OCR/transliteration and MT, yielding a seamless toolchain for Assyriologists. Finally, applying this comparative lens to other under-documented Semitic and ancient languages will test the generality of "pre-train on related language + fine-tune" and few-shot paradigms across diverse scripts, dialects, and time periods. By pursuing these threads, we aim to push the frontier of low-resource, historical-language translation ever closer to full academic and cultural utility.

Acknowledgments

This work has been partially supported by the CIDEXG/2023/12 project, funded by the Generalitat Valenciana.

References

Ahmed M. Agiza, Kai Zhu, Tianlong Zhang, and Shijie Han. 2024. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16196–16205.

Jeremy Black, Graham Cunningham, Eleanor Robson, and Gábor Zólyomi. 2002. The electronic text corpus of sumerian literature. https://etcsl.orinst.ox.ac.uk/index1.htm. Accessed: 2025-03-17.

CDLI Contributors. 2025. Cuneiform digital library initiative: Home. https://cdli.mpiwg-berlin.mpg.de/. Accessed: 2025-01-21.

Jarle Ebeling. 2007. The electronic text corpus of sumerian literature. *Corpora*, 2(1):111–120.

Zhengluo Fu, Guangxiang Zhou, Xiaogang Li, Jin Wang, Aiyun Zeng, Liang Lyu, and Xing Zhou. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *arXiv preprint arXiv:2304.04052*.

Candace Kaleimamoowahinekapu Galla. 2018. Digital realities of indigenous language revitalization: A look at hawaiian language technology in the modern world. *Language and Literacy*, 20(3):100–120.

Andrew George. 2007. *Babylonian and Assyrian: A history of Akkadian*, pages 31–71. British School of Archaeology in Iraq.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. *A survey of quantization methods for efficient neural network inference*, pages 291–326. Chapman and Hall/CRC.

Kirk Grayson, Jamie Novotny, and Poppy Tushingham. 2025. The royal inscriptions of assyria online (riao) project. https://oracc.museum.upenn.edu/riao/. Accessed: 2025-03-17.

Md. Arid Hasan, Prerona Tarannum, Krishno Dey, Imran Razzak, and Usman Naseem. 2024. Do large language models speak all languages equally? a comparative study in low-resource settings.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shengyuan Wang, and Weizeng Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Pratik S Joshi, Chathurika Welivita, Pubudu Liyanapathirana, Ameya Budhiraja, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Sevcik, Roman Vólner, Anthony Aue, Alexandra Birch, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Frank. A. Krueger. 2023a. I built the world's largest translated cuneiform corpus using ai. https://praeclarum.org/2023/06/09/cuneiform.html. Accessed: 2025-03-17.

Frank A. (praeclarum) Krueger. 2023b. Dataset for cuneiform language translation. Accessed [2025-03-19].

Nir Kshetri. 2024. Linguistic challenges in generative artificial intelligence: Implications for low-resource languages in the developing world. *Journal of Global Information Technology Management*, 27(2):95–99.

Arthur Lachaux, Baptiste Lucic, Thomas Kirchner, Arthur Mensch, Sander Lowenthal, Joseph Rouzé, Jean-Baptiste Mensch, Quentin Yu, Thibaut Batigne, Clara Stone, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*.

NLLB Team. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Oracc. 2019. ATF Primer. https://oracc.museum.upenn.edu/doc/help/editinginatf/primer/index.html. Accessed: 2025-03-17.

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. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D 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. Association for Computational Linguistics.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

Ravneet Punia, Niko Schenk, Christian Chiarcos, and Émilie Pagé-Perron. 2020. Towards the first machine translation system for sumerian transliterations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3454–3460. International Committee on Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Mihir 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.

Denise Schmandt-Besserat. 2014. *The Evolution of Writing*, pages 1–15. Elsevier.

Peng Shu, Junhao Chen, Zhengliang Liu, Hui Wang, Zihao Wu, Tianyang Zhong, Yiwei Li, Huaqin Zhao, Hanqi Jiang, Yi Pan, et al. 2024. Transcending language boundaries: Harnessing llms for low-resource language translation. *arXiv preprint arXiv:2411.11295*.

Michael P. Streck and Nathan Wasserman. 2025. Sources of early akkadian literature: Love literature. https://oracc.museum.upenn.edu/akklove/. Accessed: 2025-03-17.

Jianlin Su, Yu Lu, Shengfeng Xu, and Ahmed Murtadha. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*, pages 2214–2218.

Steve Tinney, Eleanor Robson, Niek Veldhuis, and Jamie Novotny. 2025. Open richly annotated cuneiform corpus. https://oracc.museum.upenn.edu/. Accessed: 2025-01-25.

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 *Advances in neural information processing systems*, volume 30.

Aiying Yang, Xiaotian Jin, Rui Zhang, Yibo Zhang, An Li, Jiaming Wu, Wenbin Wang, Chen Xu, Qian Wang, Xing Zhou, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

United We Fine-Tune: Structurally Complementary Datasets for Hope Speech Detection

Priyadharshini Krishnaraj[1], Tulio Ferreira Leite da Silva[1,2], Gonzalo Freijedo Aduna[3], Samuel Chen[2], Farah Benamara[4,5], Alda Mari[3]
[1] CNRS@CREATE LTD, Singapore
[2] University of Sao Paulo, Brazil
[3] Institut Jean Nicod CNRS/ENS/EHESS/PSL University
[4] IRIT, Université de Toulouse, CNRS, Toulouse INP, Toulouse, France
[5] IPAL, CNRS-NUS-A*STAR, Singapore

Abstract

We propose a fine-tuning strategy for English Multiclass Hope Speech Detection using Mistral, leveraging two complementary datasets: PolyHope and CDB, a new unified framework for hope speech detection. While the former provides nuanced hope-related categories such as GENERALIZED, REALISTIC, and UNREALISTIC HOPE, the later introduces linguistically grounded dimensions including COUNTERFACTUAL, DESIRE, and BELIEF. By fine-tuning Mistral on both datasets, we enable the model to capture deeper semantic representations of hope. In addition to fine-tuning, we developed advanced prompting strategies which provide interpretable, zero-shot alternatives and further inform annotation and classification designs. Our approach achieved third place in the multiclass (Macro F1=71.77) and sixth in the binary (Macro F1=85.35) settings.

1 Introduction

Hope speech detection has recently evolved into a specialized area of classification within NLP, aimed at distinguishing constructive and future-oriented statements from neutral or negative content. While several datasets have been proposed to support this task (Goldberg et al., 2009; Palakodety et al., 2020; Chakravarthi, 2020; Balouchzahi et al., 2023a,b), their annotation schemas vary widely—ranging from affective taxonomies to structurally grounded categories—making generalization across label sets a persistent challenge.

This paper investigates whether structurally divergent but semantically related taxonomies can be combined to improve model performance on multiclass hope speech detection. We focus on two English-language datasets: *PolyHope* (Balouchzahi et al., 2023b), which classifies hope expressions into affective categories (GENERALIZED, REALISTIC, and UNREALISTIC HOPE), and *CDB* (Ferreira Leite da Silva et al., 2025), which bases its classification on the semantic notion of *modality* (Kratzer, 1991; Portner, 2009) in the broad sense, as encompassing propositional attitudes and speech acts (Giannakidou and Mari, 2021, 2026), and thus encoding the propositional structure of hope-related speech (COUNTERFACTUAL, DESIRE, BELIEF). Despite having disjoint label sets, both datasets target overlapping semantic phenomena. We treat them as complementary sources of supervision and fine-tune a Mistral-7B model on the merged corpus using a parameter-efficient strategy.

Our methodology is informed by recent findings in multi-task and cross-taxonomy learning. Prior work shows that combining tasks with high structural complementarity can produce synergistic gains in generalization, a phenomenon referred to as the "cocktail effect" (Brief et al., 2024). For example, Lai et al. (2024) proposed Multi-Task Implicit Sentiment Analysis (MT-ISA) which leverages auxiliary sentiment tasks to enhance main-task performance, while Ivison et al. (2023) Data-Efficient Fine-Tuning (DEFT) which shows that structural similarity between tasks is often a more reliable indicator of transfer effectiveness than surface-level alignment or data volume. Building on these insights, we treat PolyHope and CDB not as competing annotation schemes, but as complementary lenses on the semantic domain of hope. Instead of aligning or mapping between taxonomies, we fine-tune a generative model on both datasets simultaneously, alternating prompt formats within a single training pipeline. This setup enables the model to internalize both affective and propositional representations of hope.

In addition to supervised fine-tuning, and drawing on recent surveys of prompting strategies (Schulhoff et al., 2025; Fagbohun et al., 2023; White et al., 2023), we propose three zero-shot prompting methods tailored to the PolyHope tax-

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 48–58, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-006

onomy for hope speech detection: *Confidence-Structured Output Prompting*, *Multiple Reasoning Path Prompting*, and *Decision Tree Prompting*. These strategies—developed specifically for this study—were designed to enhance model interpretability and decision consistency, particularly in multiclass scenarios where category boundaries are conceptually nuanced. By integrating structural supervision with reasoning-aware prompting, we evaluate both supervised and prompt-based approaches within a unified framework.

Our submission to the RANLP-2025 shared task ranked **3$^{\text{rd}}$** in the English multiclass classification track and **6$^{\text{th}}$** in the binary. Results suggest that cross-taxonomy fine-tuning without explicit task weighting, can yield competitive generalization performance. The codebase and prompt templates will be released publicly to support further research in structure-aware hope speech classification[1]. Our main contributions are:

1. A cross-taxonomy LLMs fine-tuning strategy leveraging structurally complementary datasets.

2. Proposal and evaluation of reasoning-aware zero-shot prompting strategies tailored to hope speech multiclass classification.

3. A qualitative error analysis of the best model, highlighting systematic confusion patterns in hope classification.

We begin in Section 2 with a review of previous work in the field. Section 3 details the datasets employed in the shared task. Our methodology is described in Section 4, and the corresponding results are reported in Section 5.

2 Related Work

2.1 Hope Speech Datasets

Prior research on hope speech has explored a range of perspectives, from peace-oriented discourse (Palakodety et al., 2020) to multilingual detection for promoting inclusion (Chakravarthi, 2020). Other works have examined expressions of regret and past-oriented hope (Balouchzahi et al., 2023a), and the expression of wish in products reviews and political discussions (Goldberg et al., 2009).

[1]https://github.com/Priyaaa-hub/Shared-task-prompts.git

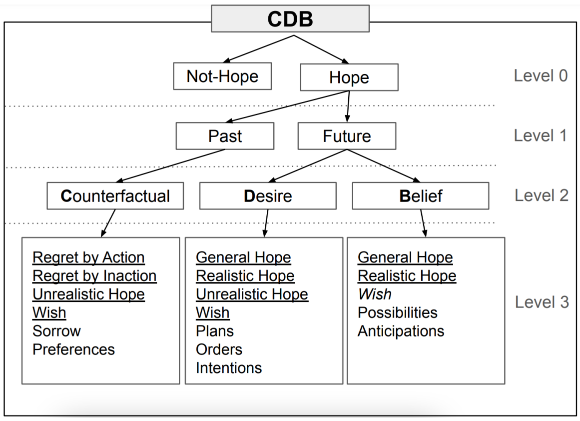


Figure 1: The COUNTERFACTUAL-DESIRE-BELIEF (CDB) model (Ferreira Leite da Silva et al., 2025).

The *PolyHope* dataset (Balouchzahi et al., 2023b) used in the shared task is annotated with four categories of future-oriented hope-related expressions: (a) NOT-HOPE, indicating the absence of hope; (b) GENERALIZED HOPE, referring to vague or non-specific statements of hope; (c) UNREALISTIC HOPE, denoting overly optimistic or implausible expectations; and (d) REALISTIC HOPE, capturing grounded and plausible expressions of hope. In contrast, the ReDDit dataset (Balouchzahi et al., 2023a) focuses exclusively on *past-oriented* hope, specifically targeting expressions of retrospective longing or regret.

Building on these prior works, the *CDB model* (Ferreira Leite da Silva et al., 2025) introduces a more fine-grained and linguistically grounded classification system. Unlike PolyHope and ReDDit, which each target a single temporal dimension of hope, the CDB model incorporates both: one class for past-oriented hope, two distinct classes for future-oriented hope, and one for the not-hope instances. This classification is grounded in the degree of speaker commitment implied by each expression, allowing for a more nuanced framework for annotation and classification. The model defines four core classes that subsume previous classification schemes, as illustrated in Figure 1: (a) NOT-HOPE: indicating the absence of any hope-related expression. (b) COUNTERFACTUAL: which captures expressions of regret and represents past-oriented hope. (c) DESIRE: encompassing future-oriented expressions of mere desire or wishful thinking that lack strong speaker commitment. (d) BELIEF: which also encodes future-oriented hope, but in this case grounded in epistemic or deontic considerations.

This taxonomy enables a more linguistically informed and temporally aware analysis of hope speech across discourse contexts. Two annotators achieved a Cohen's kappa of 74.88 for the binary classification task, and 70.46 for the multiclass classification, indicating substantial agreement given the subjectivity of the task.

2.2 Hope Speech Automatic Detection

Several shared tasks have also advanced the study of hope speech in multilingual and multicultural settings. At LT-EDI 2022 (Chakravarthi et al., 2022), LT-EDI 2023 (Kumaresan et al., 2023) and IberLEF 2023 (Jiménez-Zafra et al., 2023), the task was framed as binary classification in a variety of languages. These tasks laid the foundation for more detailed distinctions explored in subsequent years.

The IberLEF 2024 shared task (García-Baena et al., 2024) introduced a two subtasks reflecting distinct dimensions of hope: (1) *Hope for Equality, Diversity, and Inclusion* to detect supportive speech toward vulnerable groups, (2) *Hope as Expectations* that requires multi-class classification of generalized, realistic, and unrealistic expressions of future-oriented hope. Approaches based on transformer models, as well as those leveraging prompting with large language models (LLMs), have both demonstrated competitive performance. For instance, the top-ranked system in Subtask (1) (Thuy and Thin, 2024) employed a zero-shot prompting strategy using ChatGPT-3.5, incorporating class definitions into the prompt in both English and Spanish. Their solution explored multiple prompting techniques—zero-shot, one-shot, three-shot, and chain-of-thought (CoT)—combined with six different information strategies, including role-defining, class explanations, and task-specific concepts. The best performance was achieved using a one-shot prompt and an information-rich strategy that defined both class meanings and model roles, yielding a Macro F1-score of 0.7161 on the out-of-domain Spanish test set.

In Subtask (2), the winning team (Bui Hong et al., 2024) adopted a supervised approach, combining multilingual transformer models with rigorous data pre-processing and augmentation. Their method leveraged data combination across English and Spanish corpora and generated synthetic samples for minority classes using Gemini LLMs. Classification was performed via fine-tuned models such as XLM-R, mDeBERTa, and RoBERTuito, and predictions were aggregated using a max voting ensemble. This robust pipeline achieved the highest scores in the multiclass subtasks for both English (Macro F1 = 72.00) and Spanish (Macro F1 = 66.68), highlighting the effectiveness of multilingual augmentation and ensemble-based inference.

As we can see, the progression of hope speech detection methods—from traditional machine learning models to transformer architectures and, more recently, to prompt-based large language models—reflects a broader shift in NLP toward more flexible and powerful approaches, particularly for multilingual and cross-domain applications.

3 Datasets

We rely on two datasets, PolyHope and CDB. We first present them, then explain their complementarity.

3.1 PolyHope Dataset

The dataset consists of **8,256 tweets** collected in 2022, covering topics such as abortion rights, racial justice, religion, and politics. As illustrated in Table 2, the dataset exhibits moderate imbalance, with category NOT-HOPE comprising nearly half of the instances, while the remaining categories are notably less represented (GENERALIZED HOPE being more than twice as frequent as REALISTIC HOPE and nearly three times as frequent as UNREALISTIC HOPE). As the test set was not provided, we only report the statistics of the train and dev sets in the tables.

3.2 The CDB Dataset

The CDB dataset comprises **4,370 texts** in total, of which 3,092 were randomly selected and re-annotated from existing corpora (WISH (Goldberg et al., 2009), PolyHope (Balouchzahi et al., 2023b), and HopeEDI (Chakravarthi, 2020)), and 1,278 were newly collected from X (formerly Twitter) and Reddit (HopeDrone). As shown in Table 3, the dataset exhibits a slight class imbalance in the binary setting, with the HOPE category accounting for 57.28% of the total instances. In the multiclass setting, however, the distribution is more skewed: while NOT-HOPE remains the largest single class, the DESIRE category represents nearly one-third of the dataset, followed by BELIEF at just over one-fifth. The COUNTERFACTUAL category is notably underrepresented, comprising less than 5% of all texts. These proportions remain consistent across

Category	HopeDrone	PolyHope	WISH Corpus	HopeEDI	Total
COUNTERFACTUAL	15 (0.34%)	112 (2.56%)	62 (1.42%)	14 (0.32%)	**203 (4.65%)**
DESIRE	224 (5.13%)	682 (15.60%)	360 (8.24%)	113 (2.59%)	**1,379 (31.56%)**
BELIEF	390 (8.92%)	202 (4.62%)	226 (5.17%)	103 (2.36%)	**921 (21.08%)**
NOT-HOPE	649 (14.85%)	279 (6.39%)	569 (13.02%)	370 (8.47%)	**1,867 (42.72%)**
Total	**1,278 (29.24%)**	**1,275 (29.18%)**	**1,217 (27.85%)**	**600 (13.73%)**	**4,370 (100%)**

Table 1: The CDB dataset, following Ferreira Leite da Silva et al. (2025). The "Total" column aggregates the instance counts across all four datasets—HopeDrone, PolyHope, WISH Corpus, and HopeEDI—for each class in the CDB taxonomy. The bottom row summarizes the total number and relative size of each dataset.

Binary	**Train**	**Dev**	**Total**
NOT HOPE	2,245 (49.44%)	816 (49.45%)	3,061 (49.44%)
HOPE	2,296 (50.56%)	834 (50.55%)	3,130 (50.56%)
Multiclass	**Train**	**Dev**	**Total**
NOT HOPE	2,245 (49.44%)	816 (49.45%)	3,061 (49.44%)
GENERALIZED HOPE	1,284 (28.28%)	467 (28.30%)	1,751 (28.28%)
REALISTIC HOPE	540 (11.89%)	196 (11.88%)	736 (11.89%)
UNREALISTIC HOPE	472 (10.39%)	171 (10.36%)	643 (10.39%)

Table 2: Distribution of classes for binary and multi-class settings (PolyHope) by Split. The "Total" column presents the aggregate number of instances for each class, obtained by summing the respective values from the training and development splits.

the training and test splits.

Importantly, we verified that **1,020 texts** in the CDB dataset were originally drawn from the PolyHope corpus used in the shared task—**543** from the training set, **203** from the development set, and **274** from the test set. This overlap is explicitly reported to ensure transparency. During fine-tuning, the Mistral model was trained jointly on the PolyHope and CDB training sets. Crucially, the PolyHope test set remained unlabeled and was never used during training. Although some texts may have been seen with alternative annotations from the CDB taxonomy, their original PolyHope labels were hidden throughout. Rather than constituting test contamination, this setup enables robust cross-taxonomy learning and allows the model to internalize divergent labeling schemes over shared inputs.

Binary Label	**Train**	**Test**	**Total**
NOT-HOPE	1,599 (43.03%)	268 (40.98%)	1,867 (42.72%)
HOPE	2,117 (56.97%)	386 (59.02%)	2,503 (57.28%)
Multiclass Label	**Train**	**Test**	**Total**
NOT-HOPE	1,599 (43.03%)	268 (40.98%)	1,867 (42.72%)
DESIRE	1,149 (30.92%)	230 (35.17%)	1379 (31.56%)
BELIEF	801 (21.56%)	120 (18.35%)	921 (21.08%)
COUNTERFACTUAL	167 (4.49%)	36 (5.50%)	203 (4.65%)

Table 3: Distributions of classes for binary and multi-class settings (CDB model) by Split.

3.3 Cross-Taxonomy Comparison

Figures 2 and 3 illustrate the cross-taxonomy relationship between PolyHope and CDB, as reported in (Ferreira Leite da Silva et al., 2025). Each figure presents a correlation matrix based on randomly selected 1,022 PolyHope instances that were re-annotated using the CDB schema.

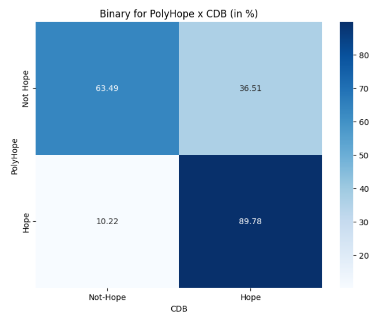


Figure 2: CDB vs. PolyHope binary annotations.

As previously stated, these datasets are not competing but complementary. Despite what Figure 2 may initially suggest—many instances labeled as NOT HOPE in PolyHope are reclassified as HOPE in CDB—the multiclass perspective reveals their compatibility. In Figure 3, we observe that the CDB category DESIRE serves as a good approximation for all three hope categories in PolyHope.

The fact that most PolyHope instances labeled as GENERALIZED, REALISTIC, or UNREALISTIC hope are mapped to the DESIRE category in CDB highlights a key difference between the taxonomies: CDB places greater emphasis on temporal and modal structure, while PolyHope focuses on plausibility and affective nuance.

This also reflects, among other factors, a structural divergence—PolyHope excludes past-oriented hope from its schema, whereas CDB explicitly encodes it through the COUNTERFACTUAL category. The divergence becomes even more apparent in the multiclass comparison (cf. Figure 3),

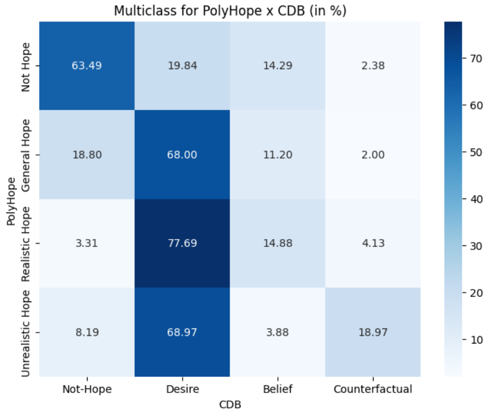


Figure 3: CDB vs. PolyHope multiclass annotations.

where there is no one-to-one mapping between categories across the two taxonomies.

We provide below some examples of PolyHope and CDB annotations. Instances (1) and (2) show cases where utterances labeled as NOT HOPE in PolyHope were annotated as BELIEF in CDB:

(1) "**Don't expect Chris Rock to talk** about 'the slap' when he performs Friday at the AVA at Casino del Sol."

(2) "Getting on your knees to pray **should stay off** the football field and stay in church settings."

Similarly, utterances (3) and (4) labeled as NOT HOPE in PolyHope, were classified as COUNTERFACTUAL in CDB, due to the presence of past-tense constructions:

(3) "Thank you for being brave and speaking up—your work is so beautiful! **Wish we had met** in NFT NYC."

(4) "I **wish it was** Speak Now, but the signs seem to point to 1989."

This structural mismatch has direct implications for training strategies. While the multiclass setup benefits from the complementarity of both taxonomies, the binary classification reveals overlapping boundaries that risk making the label space more concorrente than complementary. We therefore did not adopt the multi-task learning in the binary setting, as merging categories could introduce conflicting signals during training. Conversely, in the multiclass scenario, the alignment between affective subtypes and temporal-motivational roles supports more synergistic learning.

4 Methodology

4.1 Models

We designed models based on transformers and Large Language Models (LLMs). The reported scores are averaged over 3 runs on the PolyHope development set, as the test set has not been released. The hyper-parameters used for fine-tuning both the LLMs and transformer based models are available in the supplementary material associated to this submission.

4.1.1 Transformers

We use BERT (Bidirectional Encoder Representations from Transformers) as a baseline transformer model for hope speech classification. Known for its strong performance across a variety of NLP tasks, BERT was fine-tuned separately for both binary and multiclass classification.

4.1.2 Large Language Models

We make use of two models:

–**GPT-4**, the Generative Pretrained Transformer 4 developed by OpenAI, was also explored in our experiments. Although its performance on the development dataset was comparatively lower, it was still used to generate predictions under a zero-shot prompting strategy for the test set and included in the final submission.

–**Mistral FT**. It is the Mistral-7B-Instruct-v0.3, a 7-billion parameter open-weight LLM developed by Mistral AI, optimized for instruction-following tasks. We fine-tune Mistral using a parameter-efficient strategy based on QLoRA, a lightweight variant of Low-Rank Adaptation (LoRA) that enables scalable tuning of large language models with limited compute. This setup allows us to fine-tune Mistral-7B on structurally complementary datasets (PolyHope and CDB) while preserving efficiency and reducing overfitting. Unlike prior work comparing multiple adaptation methods, our goal is not to benchmark fine-tuning techniques, but rather to validate the value of combining taxonomically divergent supervision sources. During training, each instance included three fields: a dataset-specific prompt, the corresponding input text, and the gold classification label.

The set of prompting strategies used to fine-tune Mistral are as follows:

1. **Zero-Shot Prompting:** The prompt includes a description of the task, definitions for each label, and the expected output format, without providing any training examples.

2. **Few-Shot Prompting:** In addition to the task description and label definitions, this prompt includes two randomly chosen examples per class. There are 4 examples for binary classification (2 - HOPE and NOT HOPE), and 8 examples for multiclass classification (2 per class resp.). Each example is paraphrased before including it in the prompts to prevent overfitting and ensure the model learns from the examples and does not rely on the specific phrasing. We use a tool called quillbot[2] for paraphrasing the chosen examples after which these examples are included in the prompt for classification.

3. **Decision Tree Prompting:** This strategy implements a logical flowchart in prompt form, requiring the model to follow a step-by-step decision process. We pose several sequential queries to the model, inquiring about the presence of hope, any specific goal mentioned, and the feasibility of the goal by asking whether it involves a particular result and whether they are attainable or not. This prompt was elaborated based on the concept of *Decision Tree Reasoning for Prompting*, proposed as a structured decomposition strategy within the Tree-of-Thought framework by Yao et al. (2023), and categorized under Logical and Sequential Processing in (Fagbohun et al., 2023). It expands on classic Chain-of-Thought prompting by introducing a branching logic format for inference.

4. **Confidence-Structured Output Prompting:** This technique generates both a classification label and a graded confidence score for each category, based on observable features of the input. The prompt structure guides the model through identifying hope-related language, assessing the specificity of the desired outcome, and evaluating its feasibility. These components are followed by a confidence estimate for each label and the final classification. The method is inspired by uncertainty-aware reasoning and structured prediction techniques in LLMs. This prompt was elaborated based on the concept of *Uncertainty-Routed Chain-of-Thought (CoT)* prompting, proposed in (Schulhoff et al., 2025), and classified under Thought Generation and Self-Criticism strategies. It leverages confidence estimation techniques to refine final predictions.

5. **Multiple Reasoning Path Prompting:** This method encourages LLMs to perform a multi-perspective analysis of the text by decomposing the reasoning process into three steps: linguistic cues, goal assessment, and contextual framing. Each perspective contributes to the final classification. Such an approach is related to multi-perspective CoT prompting. This prompt was elaborated based on the concept of *Multi-View or Multi-Faceted Reasoning*, explicitly discussed in (Schulhoff et al., 2025) under Contrastive CoT and Meta-CoT, and structurally aligned with the Cognitive Verifier pattern in (White et al., 2023), which decomposes reasoning into modular sub-analyses to enhance robustness and explanatory power.

[2]https://quillbot.com/paraphrasing-tool

The prompts used for LLMs and the hyper-parameters used for fine-tuning both the LLMs and transformer-based models, are provided in the Appendix.

4.2 Submitted Systems

A total of 9 systems have been submitted for the shared task:

1. **Binary Classification:** GPT-4, Mistral FT (Zero-shot$_P$, Few-shot$_P$, Confidence Score$_P$, Multiple Reasoning$_P$).

2. **Multiclass Classification:** BERT$_P$, GPT-4, Mistral FT (Zero-shot$_P$, Zero-shot$_{P+CDB}$)

Where $Prompt_d$ indicates **Mistral FT** model fine-tuned with one of the previous 5 $Prompt$ on the dataset $d \in \{P, \ P + CDB\}$. **BERT**$_P$ has only been fine-tuned on PolyHope, **GPT4**, being prompted in a zero-shot fashion. These configurations were selected based on their superior performance on the development set (see next Section).

5 Results

Table 4 presents the results for the binary classification, best scores are in bold font. We observe that most Mistral FT variants achieve relatively stable

	Development Set				Test Set			
Model	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**	**Acc**
GPT-4	77.78	76.83	76.55	76.73	53.00	52.02	51.87	77.00
Mistral FT Variants								
Zero-shot$_P$	**84.19**	**84.17**	**84.18**	84.18	85.06	84.85	84.97	84.98
Few-shot$_P$	83.08	83.01	83.01	83.03	**85.44**	**85.34**	**85.35**	85.37
Confidence Score$_P$	83.66	83.62	83.63	83.64	85.30	85.25	85.26	85.27
Multiple Reasoning$_P$	84.03	83.98	83.99	84.00	85.01	84.91	84.92	84.93

Table 4: Performances of binary classification in development vs. test sets in terms of macro P, R and F1 scores.

	Development Set				Test Set			
Model	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**	**Acc**
BERT$_P$	**71.01**	66.58	68.24	74.30	**73.31**	69.28	70.78	77.14
GPT-4	53.55	47.62	42.55	56.67	55.80	49.54	44.87	57.86
Mistral FT Variants								
Zero-shot$_P$	68.12	68.49	68.07	74.00	68.66	69.97	69.12	75.06
Zero-shot$_{P+CDB}$	70.40	**69.98**	**70.12**	75.58	71.19	**71.09**	**71.11**	76.80

Table 5: Multiclass classification results in the development vs. test sets in terms of macro P, R, and F1 scores.

Binary				Multiclass			
Label	**P**	**R**	**F1**	**Label**	**P**	**R**	**F1**
BERT$_P$							
HOPE	80.00	**86.81**	83.27	GEN. HOPE	64.86	76.66	70.26
NOT HOPE	**85.23**	77.82	81.36	REAL. HOPE	69.33	53.06	60.12
				UNREAL. HOPE	66.91	54.39	60.0
				NOT HOPE	82.94	82.23	82.58
GPT-4$_P$							
HOPE	83.48	67.27	74.50	GEN. HOPE	76.81	11.35	19.78
NOT HOPE	72.09	86.40	78.60	REAL. HOPE	31.27	64.29	42.07
				UNREAL. HOPE	39.02	28.07	32.65
				NOT HOPE	67.11	**86.76**	75.68
Mistral FT							
Zero-shot$_P$				Zero-shot$_P$			
HOPE	83.75	85.25	**84.49**	GEN. HOPE	72.25	61.88	66.67
NOT HOPE	84.64	83.09	83.86	REAL. HOPE	53.62	64.29	58.47
				UNREAL. HOPE	64.24	61.99	63.10
				NOT HOPE	82.35	85.78	84.03
Few-shot$_P$				Zero-shot$_{P+CDB}$			
HOPE	81.99	85.13	83.53	GEN. HOPE	70.45	66.38	68.36
NOT HOPE	84.18	80.88	82.50	REAL. HOPE	59.62	64.80	62.10
				UNREAL. HOPE	67.72	62.57	65.05
				NOT HOPE	**83.79**	86.15	**84.95**
Confidence-score$_P$				Confidence-score$_P$			
HOPE	82.94	85.13	84.02	GEN. HOPE	52.38	30.62	38.65
NOT HOPE	84.38	82.11	83.23	REAL. HOPE	19.45	54.59	28.69
				UNREAL. HOPE	24.64	39.77	30.43
				NOT HOPE	76.89	19.98	31.71
Multiple Reasoning$_P$				Multiple Reasoning$_P$			
HOPE	83.29	85.49	84.38	Gen. Hope	37.87	63.81	47.53
NOT HOPE	84.76	82.48	83.60	REAL. HOPE	24.41	31.63	27.56
				UNREAL. HOPE	33.33	1.75	3.33
				NOT HOPE	70.28	51.59	59.51

Table 6: Performances per class in the development set in both binary and multiclass settings.

performance across all prompting strategies, with only minor variations, outperforming GPT4. The highest performance is observed using Few-shot prompting trained on the PolyHope dataset.

Table 5 shows the results for the multiclass classification task, where Mistral has been fine-tuned either on PolyHope or PolyHope+CDB. The $Prompt_{P+CDB}$ setups consistently outperform $Prompt_P$ by approximately 2%, indicating that incorporating the additional CDB data enhances the model's capacity for fine-grained classification.

We finally provide in Table 6 per-class performance results for both the binary and multiclass classification tasks on the development set, as the testset has not been released. It offers a comprehensive overview of all evaluated strategies.

Overall, our system achieved a **6th** place ranking in the Binary Classification task and a **3rd** place in the Multiclass Classification task on the English dataset. These results confirm the effectiveness of our tailored prompting and fine-tuning strategies, particularly for multiclass scenarios.

6 Error Analysis

Here we analyze the most frequent misclassifications—**377 instances**—as predicted by our best model Mistral FT using Zero-Shot$_{P+CDB}$ on the development set. These errors can be grouped into two main categories:

– Generalized Hope (Gold) vs. Not Hope (Prediction)

1. GENERALIZED HOPE (Gold) → NOT HOPE (Prediction) = 90 instances, as in "This is awful. **Please pray for these poor people**. No one should have died that way, but will this administration do anything? Nope, they have a clown tribunal to attend to, and a constitution to ignore".

2. NOT HOPE (Gold) → GENERALIZED HOPE (Prediction) = 81 instances, as in "All task done [...] thank you and **wish** me luck".

These confusions suggest that the model struggles with vague or subtle expressions of hope (highlighted in bold in the examples). In (a), for instance, short hopeful spans are embedded in longer neutral or non-hopeful content, which may dominate the model's representation. Conversely, in (b), lexical cues like *hope* or *wish* lead to overgeneralization.

– REALISTIC HOPE (Gold) vs. GENERALIZED HOPE (Prediction)

1. GENERALIZED HOPE (Gold) → REALISTIC HOPE (Prediction) = 53 instances, like in "I just hope my 3 years of Spanish lessons and streak are still there".

2. REALISTIC HOPE (Gold) → GENERALIZED HOPE (Prediction) = 34 instances, e.g., "Well I hope we're singing Turn Out the Lights the Party's Over, when this hearing is done."

In these last two cases, the model may struggle to distinguish between grounded, outcome-oriented hopes and more diffuse or emotive expressions, suggesting limited sensitivity to contextual or pragmatic features that signal speaker intent.

7 Conclusion

We proposed novel prompting strategies that achieved top-tier performance in the shared task. In addition, our fine-tuning methodology demonstrates the feasibility of combining structurally distinct datasets—each with its own label taxonomy—for multiclass classification using large language models and transformer architectures. This cross-taxonomy approach enables richer supervision and improved generalization.

In the future, we plan to consider the idea of unifying hope speech taxonomies via latent label modeling or joint annotation projection. This could offer a principled way to formalize cross-taxonomy alignment.

Acknowledgment

This work has been supported by DesCartes: the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. Alda Mari gratefully thanks ANR-17-EURE-0017 FrontCog. Tulio Ferreira Leite da Silva is thankful to Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) 23/1010-1 for their support.

Limitations

The prompting strategies we explored such as decision tree prompting, confidence-structured output prompting, and multiple reasoning path prompting-were selected precisely for their cross-domain generalization, as highlighted in recent work (e.g., (Schulhoff et al., 2025; White et al., 2023)). However, we acknowledge that task-specific adaptation is often necessary to fully leverage their benefits.

For instance, the structural logic of decision tree prompting can be transferred across tasks, but the branching criteria must be adapted to the domain's ontology. Similarly, while the general idea behind confidence-structured output prompting is domain-agnostic (e.g., eliciting outputs with associated self-assesses certainty), the format and calibration of confidence levels might require tuning. In multiple reasoning path prompting, the principle of diverse inference paths remain reusable, but the types of reasoning paths must reflect the target task's cognitive demands. In short, while the strategies are reusable at a conceptual level, they often require lightweight, task-aware instantiations to reach optimal performance.

References

Fazlourrahman Balouchzahi, Sabur Butt, Grigori Sidorov, and Alexander Gelbukh. 2023a. ReDDIT: Regret detection and domain identification from text. *Expert Systems with Applications*, 225:120099.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2023b. PolyHope: Two-level hope speech detection from tweets. *Expert Systems with Applications*, 225:120078.

Meni Brief, Oded Ovadia, Gil Shenderovitz, Noga Ben Yoash, Rachel Lemberg, and Eitam Sheetrit. 2024. Mixing It Up: The Cocktail Effect of Multi-Task Fine-Tuning on LLM Performance – A Case Study in Finance. ArXiv:2410.01109 [cs].

Son Bui Hong, Quan Le Minh, and Van Thin Dang. 2024. ABCD team at HOPE 2024: Hope detection with BERTology models and data augmentation. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024)*, CEUR Workshop Proceedings. CEUR-WS.org.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A Multilingual Hope Speech Detection Dataset for Equality, Diversity, and Inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, Subalalitha Cn, John McCrae, Miguel Ángel García, Salud María Jiménez-Zafra, Rafael Valencia-García, Prasanna Kumaresan, Rahul Ponnusamy, Daniel García-Baena, and José García-Díaz. 2022. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 378–388, Dublin, Ireland. Association for Computational Linguistics.

Oluwole Fagbohun, Rachel Harrison, and Anton Dereventsov. 2023. An Empirical Categorization of Prompting Techniques for Large Language Models: A Practitioner's Guide. *Journal of Artificial Intelligence, Machine Learning and Data Science*, 1:1–11.

Daniel García-Baena, Fazlourrahman Balouchzahi, Sabur Butt, Miguel Ángel García-Cumbreras, Atnafu Lambebo Tonja, José Antonio García-Díaz, Selen Bozkurt, Bharathi Raja Chakravarthi, Hector G. Ceballos, Rafael Valencia-García, Grigori Sidorov, L. Alfonso Ureña-López, Alexander Gelbukh, and Salud María Jiménez-Zafra. 2024. Overview of HOPE at IberLEF 2024: Approaching Hope Speech Detection in Social Media from Two Perspectives, for Equality, Diversity and Inclusion and as Expectations. *Procesamiento del Lenguaje Natural*, 73(0):407–419. Number: 0.

Anastasia Giannakidou and Alda Mari. 2021. *Truth and veridicality in grammar and thought: Mood, modality, and propositional attitudes*. University of Chicago Press.

Anastasia Giannakidou and Alda Mari. 2026. *Modal Sentences*. Cambridge University Press.

Andrew B. Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May All Your Wishes Come True: A Study of Wishes and How to Recognize Them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271, Boulder, Colorado. Association for Computational Linguistics.

Hamish Ivison, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. 2023. Data-efficient finetuning using cross-task nearest neighbors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9036–9061, Toronto, Canada. Association for Computational Linguistics.

Salud María Jiménez-Zafra, Miguel Ángel Garcia-Cumbreras, Daniel García-Baena, José Antonio Garcia-Díaz, Bharathi Raja Chakravarthi, Rafael Valencia-García, and Luis Alfonso Ureña-López. 2023. Overview of HOPE at IberLEF 2023: Multilingual Hope Speech Detection. *Procesamiento del Lenguaje Natural*, pages 371–381.

Angelika Kratzer. 1991. Modality. *In von Stechow, A. and Wunderlich, D., editors, Semantics: An international handbook of contemporary research.*

Prasanna Kumar Kumaresan, Bharathi Raja Chakravarthi, Subalalitha Cn, Miguel Ángel García-Cumbreras, Salud María Jiménez Zafra, José Antonio García-Díaz, Rafael Valencia-García, Momchil Hardalov, Ivan Koychev, Preslav Nakov, Daniel García-Baena, and Kishore Kumar Ponnusamy. 2023. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 47–53, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Wenna Lai, Haoran Xie, Guandong Xu, and Qing Li. 2024. Multi-task learning with llms for implicit sentiment analysis: Data-level and task-level automatic weight learning.

Shriphani Palakodety, Ashiqur R. KhudaBukhsh, and Jaime G. Carbonell. 2020. Hope Speech Detection: A Computational Analysis of the Voice of Peace. ArXiv:1909.12940 [cs].

Paul Portner. 2009. *Modality*. OUP Oxford.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. 2025. The prompt report: A systematic survey of prompt engineering techniques.

Tulio Ferreira Leite da Silva, Gonzalo Freijedo Aduna, Farah Benamara, Alda Mari, Zongmin Li, Li Yue, and Jian Su. 2025. CDB: A Unified Framework for Hope Speech Detection Through Counterfactual, Desire and Belief. In *The 2025 Annual Conference of the Nations of the Americas Chapter of the ACL*.

Nguyen Thi Thuy and Dang Van Thin. 2024. An empirical study of prompt engineering with large language models for hope detection in english and spanish. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024)*, CEUR Workshop Proceedings. CEUR-WS.org.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

A Model Hyper-parameters

Tables 7 and 8 present the hyperparameter used to fine-tune the transformer-based models and large language models in our experiments.

Parameter	**Value**
Pre-trained Model	BERT-base-uncased
Max Sequence Length	256
Batch Size	16
Learning Rate	2×10^{-5}
Optimizer	AdamW
Number of Epochs	10

Table 7: Hyperparameter for BERT fine-tuning.

Parameter	**Value**
Sequence Length	2048
Gradient Accumulation	2
Learning Rate	2×10^{-5}
Scheduler	Cosine
Number of Epochs	3
Lora Rank (r)	8
Save Checkpoints	Every 1000 steps

Table 8: Hyperparameter for fine-tuning Mistral.

B Prompt Design and Examples

B.1 Prompt Structure

Figures 4, 5, 6, 7, and 8 illustrate the various prompting strategies applied in assessing the large language model.

Your Job is to classify user sentences using only **ONE** of the following categories. Please generate ONLY ONE WORD classification:
Categories:
Generalized Hope: A broad sense of optimism not tied to specific outcomes.
Realistic Hope: An expectation of optimism not tied to specific outcomes.
Unrealistic Hope: A desire for outcomes that are unlikely or impossible
Not Hope: Texts that do not express hope

Text: “{text}”
CLASSIFICATION:

Figure 4: Zero-Shot (Multiclass).

B.2 Dataset Prompt Examples

Figures 9 and 10 illustrate the prompt example instances from the PolyHope and CDB datasets, including the prompt, input text, and corresponding gold classification labels.

Your Job is to classify user sentences using only **ONE** of the following categories. Please generate ONLY ONE WORD classification:
Categories:
Hope: Any expression of optimism, positive expectation, or belief in a desired outcome. This includes:
a) A broad sense of optimism not tied to specific outcomes.
b) An expectation that is grounded in achievable goals.
c) A desire for outcomes that are unlikely or impossible.
Not Hope: Texts that do not express hope.
Examples …..
Text: "{text}"
CLASSIFICATION:

Figure 5: Few-Shot (Binary).

You are performing a hope speech classification task using a decision tree approach. For each given text, follow this decision process:

1) Does the text express any form of hope, optimism, or positive expectation?
- If NO → Classify as "Not Hope"
2) If YES, does the text express a broad sense of optimism not tied to specific outcomes?
- If YES → Classify as "Generalized Hope"
- If NO, proceed to the next step.
3) If YES, is the text grounded in achievable goals and expectations?
- If YES → Classify as "Realistic Hope"
- If NO, proceed to the next step.
4) If YES, does the text express a desire for outcomes that are unlikely or impossible?
- If YES → Classify as "Unrealistic Hope"

Analyze the input text and provide only the classification.

Examples….

Output Format:
Text: "{text}"
CLASSIFICATION:

Figure 6: Decision Tree (Multiclass).

You are performing a hope speech classification task based on multiple reasoning. Analyze this text for expressions and classify user sentences using only **ONE** of the following categories.
* Linguistic Features: What hope-related words or phrases appear in the text? What is the general sentiment? (not more than 50 words)
* Goal Assessment: Are specific outcomes mentioned? How feasible are these outcomes? (not more than 50 words)
* Context Evaluation: What is the broader context of the hope expression? Are there conditional elements to the hope? (not more than 50 words)

Categories:
* "Generalized Hope": A broad sense of optimism not tied to specific outcomes.
* "Realistic Hope": An expectation grounded in achievable goals.
* "Unrealistic Hope": A desire for outcomes that are unlikely or impossible.
* "Not Hope": Texts that do not express hope.

Examples….

Output Format:
Text: "{text}"
Linguistic Features:
Goal Assessment:
Context Evaluation:
CLASSIFICATION:

Figure 7: Multiple Reasoning (Multiclass).

You are performing a hope speech classification task based on confidence structured output. Analyze this text for expressions and classify user sentences using only **ONE** of the following categories.

Output Format:
* Hope-related elements: List any words, phrases, or sentiments related to hope
* Outcome Specificity: Are specific outcomes mentioned? What are they?
* Feasibility Assessment: If outcomes are mentioned, how feasible are they?
* Classification Confidence: Generalized Hope[0-100%], Realistic Hope[0-100%], Unrealistic Hope[0-100%], Not Hope: [0-100%]

Categories:
* "Generalized Hope": A broad sense of optimism not tied to specific outcomes.
* "Realistic Hope": An expectation grounded in achievable goals.
* "Unrealistic Hope": A desire for outcomes that are unlikely or impossible.
* "Not Hope": Texts that do not express hope.

Examples….

Output Format:
Text: "{text}"
Hope-related elements:
Outcome Specificity:
Feasibility Assessment:
Classification Confidence:
CLASSIFICATION:

Figure 8: Confidence Score (Multiclass).

You will classify user-provided sentences into ONE of the following categories. Output Format: Provide ONLY ONE WORD for the classification. Do not include any explanations or introductory phrases.

Categories:
- Generalized Hope: A broad sense of optimism not tied to specific outcomes.
- Realistic Hope: An expectation grounded in achievable goals.
- Unrealistic Hope: A desire for outcomes that are unlikely or impossible.
- Not Hope: Texts that do not express hope.

Text: "Well the club season is over. We ended up tied for 44th place out of 130 in the Aspire division at the AAU National Championships. I had a great time! Played well. On to Utica High School volleyball bright and early tomorrow. Go U!"
CLASSIFICATION: Generalized Hope

Figure 9: Prompt, input instance, and gold classification in the PolyHope dataset.

You will be presented with user sentences, and you must classify each sentence into one of the following categories:

- Desire: The expression is future-oriented and indicates a wish, intention, or command.
- Belief: The expression is future-oriented and indicates a belief or prediction, often epistemic or deontic.
- Counterfactual: The expression is past-oriented and implies a desire for an alternative outcome.
- Not Hope: Any sentence that does not match the above categories.

Text: "Democrats would literally say anything for more votes."
CLASSIFICATION: Belief

Figure 10: Prompt, input instance, and gold classification in the CDB dataset.

Does Anaphora Resolution Improve LLM Fine-Tuning for Summarisation?

Yi-Chun Lo[1] and **Ruslan Mitkov**[2]
[1]Lancaster University, United Kingdom
[2]University of Alicante, Spain
y.lo4@lancaster.ac.uk ruslan.mitkov@ua.es

Abstract

This study investigates whether adding anaphora resolution as a preprocessing step before fine-tuning the text summarisation application in Large Language Model (LLM) can improve the quality of summary output. We conducted two sets of training with the T5-base model and BART-large model using the SAMSum dataset. One used the original text and the other used the text processed by a simplified version of MARS (Mitkov's Anaphora Resolution System). The experiment revealed that when T5-base model was fine-tuned on the anaphora-resolved inputs, the ROUGE-1, ROUGE-2 and ROUGE-L metrics were improved from 45.8567, 22.0195 and 38.0433 to 48.0281, 24.4447 and 40.3584 respectively (Wilcoxon signed-rank test p-value less than 0.01 and paired t-test p-value less than 0.01). In contrast, BART-large model only had a slight improvement after fine-tuning under the same conditions, which was not statistically significant. Further analysis of the generated summaries confirmed that anaphora resolution was helpful in semantic alignment. In conclusion, this study demonstrates that adopting anaphora resolution as a preprocessing step for LLM fine-tuning is effective in enhancing the performance of summarisation in T5-base model. Although it did not reach statistical significance on BART-large, it still has practical value for small LLM or scenarios with limited computing resources.

1 Introduction

In recent years, the rapid development of Large Language Model (LLM) has greatly contributed to the advancement of various areas in Natural Language Processing (NLP). With the increasing ability of these models to understand and generate language, text summarisation is an important and widely used application that increasingly relies on LLM for processing. Whether it is news summarisation, meeting record organisation, or social media content compression, LLM has demonstrated a strong ability to generate summaries (Gusev, 2020; Pan et al., 2024; Blekanov et al., 2022).

To further improve the performance of LLM on specific tasks, fine-tuning is one of the most common strategies. By fine-tuning on the downstream task dataset, the model can better adapt to the target task and improve the quality of the output. However, the effect of fine-tuning depends not only on the model structure design and training arguments, but also on the characteristics of the input data. In this background, anaphora resolution is particularly important. It refers to the automatic identification of the antecedent to which an expression (such as a pronoun or a noun phrase) in a text refers, and is an essential part of language interpretation. As Mitkov (2002) pointed out, anaphora resolution is a vital task for computers to comprehend natural language. Nevertheless, most of the past studies have focused on the internal evaluation of the anaphora resolution itself or analysing its overall impact on specific applications. Mitkov et al. (2007, 2012) have also investigated the results of anaphora resolution and coreference resolution (not only backward-pointing references but includes all mentions referring to the same entity) in NLP applications, and indicated that anaphora resolution can bring some degree of performance improvement. However, there is no systematic study to explore whether anaphora resolution as a data preprocessing step can significantly improve the fine-tuning effect of LLM. Therefore, in this paper, we conduct experiments aiming at the following core question:

> Can anaphora resolution preprocessing improve LLM summarisation fine-tuning?

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 59–66, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-007

2 Related Work

Before understanding anaphora resolution, it is crucial to clarify the basic concept of *anaphor*, which is a word or phrase that points back to a previous reference in a discourse, such as a personal pronoun (he, she, it) or a definite noun phrase. In contrast, antecedent is the previous entity referenced by anaphor, usually in noun phrase (NP). Take the sentence mentioned by Mitkov (2022) as an example:

> *The Queen* said the UK will succeed in its fight against the coronavirus pandemic, in a rallying message to the nation. *She* thanked people for following government rules to stay at home.

In this case, *She* is the anaphor and *the Queen* is the antecedent, which establishes semantic relationship in the discourse. Anaphora resolution is the process for identifying the antecedent of an anaphor. Among some early approaches, Lappin and Leass (1994) developed an algorithm based on syntactic structures and heuristic rules that effectively combines semantic and discourse information for anaphora resolution. Ge et al. (1998) introduced a statistical approach to the construction of anaphora resolution decision tree using a data-driven method.

Mitkov (1998); Mitkov et al. (2002) proposed a different approach to knowledge-poor anaphora resolution. This method was later evolved into MARS (Mitkov's Anaphora Resolution System), which is a fully automated system for anaphora resolution. MARS has the advantage of simplicity, fast operation, and the ability to achieve about 60% accuracy in technical manuals without relying on knowledge bases.

In addition to discourse-level preprocessing techniques, modern text summarisation applications rely heavily on pre-trained LLMs. Early on, the Sequence-to-Sequence (Seq2Seq) neural network summarisation model (Sutskever et al., 2014) was developed. This model applies an 'encoder-decoder' framework to encode the entire text before generating a summary. In simple terms, the entire paragraph is encoded into a vector. The decoder then uses this vector and the generated words to generate a summary word by word. However, the vector cannot accommodate long texts, and key information from the beginning can be easily missed. Therefore, many studies have incorporated 'attention' into the encoder-decoder architecture (Bahdanau, 2014; Rush et al., 2015; Luong et al., 2015). During encoding, the state of each position is output. For each generated word, the decoder calculates a set of attention weights to focus on the most relevant positions. Subsequent research has incorporated the Transformer (Vaswani et al., 2017), using multi-head self-attention to model the entire text. In the encoder, self-attention is used to enable each word in a text to look back at other words in the text and determine which to focus on at the moment. The decoder uses masked self-attention to focus only on the generated portion, and cross-attention to allow the model to consider the most relevant parts of the original text when outputting the summary.

Among them, T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) is a representative Transformer model. In addition to the architecture mentioned above, the core concept of T5 is span corruption. During pre-training, a continuous segment of text is first removed from the original source, prompting the model to reconstruct the omitted passage. This is like asking the model to understand the context and fill in the missing content with its own words, just like the ability to read and retell the text required for summarisation. The design is not only flexible, but also allows it to perform well on a variety of summary datasets (Zhang et al., 2020; Hasan et al., 2021; Guo et al., 2021).

In contrast, BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2019) is another representative Transformer model. Unlike T5, in addition to removing consecutive segments, BART also utilises a denoising autoencoder to scramble the input before requiring the model to recover it. This is done to train the model to have greater understanding and reconstruction capabilities. This destruction-reconstruction method also enables BART to perform well on summary tasks (Yu et al., 2020; Yadav et al., 2023).

However, most studies have focused on the optimisation of the model itself, and have rarely explored the need for semantic enhancement of the input data in the fine-tuning process. Therefore, this is exactly the problem that this study aims to investigate.

3 Data

The dataset used in this study is SAMSum Corpus (Gliwa et al., 2019), a manually annotated conversation summary dataset of simulated two-person real-time chats in everyday life. There are more than 16,000 conversations in this dataset, each containing multiple rounds of speech with corresponding concise summaries. The dialogues are written and annotated by linguists, with a clear semantic structure and consistent style. The dataset is widely used in summarisation research and is one of the most common standardised assessment corpora available.

SAMSum is particularly suitable for this study due to the following reasons. Firstly, the data are multi-round spoken dialogues with a large number of pronouns, which are very likely to be ambiguous, and this is exactly the context in which anaphora resolution can be useful. Secondly, the output summaries of SAMSum are all abstractive style, so the model needs to have a deep understanding of semantics and discourse coherence in order to produce high quality summaries. By comparing the effect of fine-tuning before and after anaphora resolution, the effect of discourse clarity on model learning can be effectively observed. Although other datasets such as MeetingBank (Hu et al., 2023) and CNN/DailyMail (Nallapati et al., 2016) were also considered, most of these datasets do not have the conversational interactivity of SAMSum and do not require as much to identify antecedents in summaries. Furthermore, these datasets are larger than SAMSum. Given limited computing resources, SAMSum may be the most cost-effective choice.

However, the dataset has some limitations. As the conversations are simulated, they may not be as natural as real social platform conversations, and the scenarios are relatively focused on everyday conversations, which lacks topic diversity. Nevertheless, SAMSum is highly representative in terms of data quality, annotation consistency and task relevance, and is a suitable test to assess whether LLM benefits from discourse-level preprocessing such as anaphora resolution.

4 Methodology

The methodology of this study is divided into two stages. Firstly, anaphora resolution is performed on the dialogue texts of the training set in the SAMSum dataset using a self-implemented simplified version of MARS, in which the anaphor are replaced by their inferred antecedents. Then, T5 and BART models are fine-tuned using the anaphora resolution and the unprocessed versions of the data. Finally, by comparing the performance of the models in generating summaries on the test set, we analyse whether introducing anaphora resolution in data preprocessing can effectively improve the performance of the summarisation. In other words, we start with LLM that has been pre-trained on a large-scale corpus. To help the model learn to output summaries based on inputs, we fine-tune it on the SAMSum dataset, aligning its generated distribution with the target summaries. After training, during inference and testing, the model employs an autoregressive approach, conditioning on previously generated tokens to generate the next token. This study aims to investigate whether performing anaphora resolution on the SAMSum dataset during the fine-tuning phase can improve the final summarisation performance of the model.

4.1 Anaphora Resolution with MARS

In this study, a simplified version of MARS (Mitkov's Anaphora Resolution System) is used, with the core logic continued from the framework of Mitkov et al. (2002), which is approximately the same as its five processing phases. First, the system applies the FDG Parser from Conexor (Tapanainen and Jarvinen, 1997) to perform part-of-speech (POS) tagging, lemmatisation, and dependency parsing on the input text to extract compound NPs for subsequent use. Then, in the second stage, the system identifies potential referential pronouns and filters out non-referential 'it' by the machine learning method developed by Evans (2001). In the third stage, for each identified referential pronoun, NPs are selected as antecedent candidates from the heading of the paragraph, the current sentence and the first two sentences. Further filtering is performed according to grammatical constraints, requiring gender and number agreement between candidates and pronouns, and excluding grammatically impossible combinations. The fourth stage applies a set of antecedent indicators to all qualified candidates, which contain a total of 14 preferential and impeding factors, and each candidate receives a set of scores based on these indicators to measure its likelihood of becoming an antecedent. Finally, in the fifth stage, the candidate with the highest total score is chosen as the antecedent of the anaphor. In case of a tie, the most recent highest-scoring candidate is chosen.

However, there are many differences in the implementation details. First, in the syntactic analysis stage, considering the open source and efficiency issues, spaCy (Honnibal et al., 2020) is used to replace the original FDG Parser to perform POS tagging, dependency parsing, and to count the frequency of occurrence for NPs. In the second stage of pleonastic it filtering, the machine learning classifier proposed by Evans (2001) is abandoned and part of the discrimination rule proposed by Paice and Husk (1987) is applied instead. For the third stage of candidate extraction, the gender agreement check is omitted because of the uncertainty in the correspondence between names and genders in the conversation dataset and the high risk of gender mismatch. During the fourth stage, the original 14 indicators other than boost pronoun are employed. However, collocation match only compares the lemma without creating a collocation database, and term preference replaces the original TF-IDF method with the highest-frequency occurring NPs. In addition, instead of implementing a Genetic Algorithm (GA) for automatic weight optimisation (OrĂsan et al., 2000), the system adopts a fixed score, which is expected to run in a more stable and lighter way.

4.2 Fine-Tuning Setup

This study utilises T5-base and BART-large. T5-base is a publicly available version of the intermediate pre-training model in the T5 architecture, which has about 220M parameters with a complete encoder-decoder structure. BART-large is a high-level pre-training model based on the BART architecture, including a 12-layer encoder and a 12-layer decoder, with a total of approximately 402M parameters. These models strike a balance between resource consumption and model performance. In addition, this model selection can also take into account the variations in the scale of two parameters and test the performance of models with different structures. The original version of the SAMSum dataset has been divided into training and testing sets, so this study directly follows its default partitioning for model training and testing without any additional adjustment. We have designed two sets of inputs. One is the original dialogue data and the other is the anaphora-resolved version by MARS. Each is used to fine-tune models with the same structure and settings, so that a fair comparison can be made as to whether anaphora resolution improves model summarisation.

For the training arguments, the batch size is set to 8, the learning rate is set to 0.0001, and the training is conducted with 3 epochs. In order to retain some of the pre-training knowledge and reduce the consumption of resources, the weights of the first three encoder layers in both T5-base and BART-large are frozen. The optimiser employs AdamW (Loshchilov and Hutter, 2017) with a linear scheduler, where the learning rate decreases as the training progresses. Moreover, ROUGE-1, ROUGE-2 and ROUGE-L are considered as the summary quality assessment metrics in the test set (Lin, 2004).

In order to verify the differences in summary quality between different input versions are not due to random fluctuations, this study conducts Wilcoxon signed-rank tests (Wilcoxon, 1992) and paired Student's t-tests on the ROUGE-1, ROUGE-2, and ROUGE-L metrics of each sample in the test set. All tests are one-tailed, with the alternative hypothesis that the anaphora-resolved result increases higher ROUGE metrics. Furthermore, the Holm–Bonferroni method (Holm, 1979) is used to correct the multiple comparison results of the three metrics, with the significance level set to 0.01.

To ensure the reproducibility of our experiments, we set the number of random seeds to 413, and use the L4 GPU of Google Colab for training.

5 Results

After anaphora resolution on the SAMSum dataset, 2,479 (91.679%) of the 2,704 target pronouns were replaced. Consistent with the original MARS, the antecedent candidates in this study were restricted to the current sentence and the two preceding sentences. Of these replaced pronouns, approximately 48.81% had their antecedents in the same sentence, 31.18% in the previous sentence, and 20.01% in the previous two sentences. On average, each dialogue contained 3.3 pronouns. Anaphora resolution only slightly altered the input length, increasing each dialogue (per sample) by an average of 1.3056 tokens and 35.6174 characters. Moreover, this section reports the performance of the four fine-tuned models on the SAMSum test set in turn. First, two sets of results are presented for T5-base (original vs. resolved), and then two sets of results for BART-large (original vs. resolved).

5.1 T5-base

Table 1 lists the ROUGE metrics of the T5-base model on original input and the anaphora-resolved input. From the results, it could be seen that with the integration of anaphora resolution, the model showed significant improvement in all three ROUGE metrics. The one-tailed Wilcoxon signed-rank test (W-test) and the paired Student's t-test (t-test) results including test statistics, raw p-values, and Holm–Bonferroni adjusted p-values are reported in Tables 2, 3 and 4. All three ROUGE scores had p-values close to zero, confirming that the performance improvement brought by anaphora resolution is highly significant.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Raw	45.8567	22.0195	38.0433
Resolved	**48.0281**	**24.4447**	**40.3584**

Table 1: ROUGE comparison for T5-base

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	154033.50	127586.00	151217.50
t-test	6.31	6.04	6.08

Table 2: Test statistics for Wilcoxon signed-rank test (W-test) and paired Student's t test (t-test) on T5-base ROUGE metrics

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	0	0	0
t-test	0	0	0

Table 3: Raw p-values for Wilcoxon signed-rank test (W-test) and paired Student's t-test (t-test) on T5-base ROUGE metrics

A dialogue from the SAMSum test set further demonstrated the semantic contrast between the two models. The summaries generated from the original model were compared with those from the anaphora-resolved model, as well as the artificial reference summaries. In this dialogue, Igor expresses his workload and depression during the two weeks before leaving his job, and John gives advice and counselling. However, the summary generated by the original model only mentioned that Igor was overloaded with work and focused on the persuasion of John to 'stop thinking and start doing'. It completely ignored the frustration of Igor and the assessment of John that it was irresponsible to assign too much work during the notice period. In contrast, the model summary after anaphora resolution not only captured the 'demotivated' mood of Igor, but also correctly reflected the criticism of excessive work allocation by John.

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	0	0	0
t-test	0	0	0

Table 4: Holm–Bonferroni corrected p-values for Wilcoxon signed-rank test (W-test) and paired Student's t test (t-test) on T5-base ROUGE metrics

This allowed the generated content to take into account both emotions of Igor and opinions of John, and was closer to the dual narrative of the reference summary. The full dialogue and model outputs can be found in Appendix A.1.

5.2 BART-large

Table 5 lists the ROUGE metrics of the BART-large model on original input and the anaphora-resolved input. From the overall trend, BART-large had slightly increased in all three ROUGE metrics after anaphora resolution, indicating that the semantic consistency of the generated summary has improved. The one-tailed Wilcoxon signed-rank test (W-test) and the paired Student's t-test (t-test) results including test statistics, raw p-values, and Holm–Bonferroni adjusted p-values are reported in Tables 6, 7 and 8. However, the results indicate that these improvements are not statistically significant. The p-values of these three scores are all greater than 0.01.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Raw	49.6463	26.5392	41.9366
Resolved	**50.0213**	**26.8944**	**42.1020**

Table 5: ROUGE comparison for BART-large

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	109756.50	91295.00	111061.00
t-test	0.12	0.23	0.05

Table 6: Test statistics for Wilcoxon signed-rank test (W-test) and paired Student's t test (t-test) on BART-large ROUGE metrics

On the semantic level, the BART-large model also showed obvious differences on the same test examples in Section 5.1. The original model mentioned that John suggested Igor to do what he had to do. The model after anaphora resolution clearly conveyed the view of John that it was irresponsible to assign too much work during the notice period. The full dialogue and model outputs can be found in Appendix A.2.

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	0.3909	0.3399	0.5790
t-test	0.4520	0.4106	0.4798

Table 7: Raw p-values for Wilcoxon signed-rank test (W-test) and paired Student's t test (t-test) on BART-large ROUGE metrics

Test	ROUGE-1	ROUGE-2	ROUGE-L
W-test	1	1	1
t-test	1	1	1

Table 8: Holm–Bonferroni corrected p-values for Wilcoxon signed-rank test (W-test) and paired Student's t test (t-test) on BART-large ROUGE metrics

6 Discussion

This study confirms that adding anaphora resolution before fine-tuning can significantly improve the summary quality of the T5-base model, reaching high significance in all ROUGE metrics. For BART-large, although there was a small gain, it did not pass the significance test, indicating that its marginal benefit on large models is relatively limited.

The actual summary examples also confirmed the above results. The summary of the T5-base model without anaphora resolution only focuses on the heavy workload and ignores the emotional clues. After anaphora resolution, it can fully present the frustrated state of Igor. Although BART-large can add details about the evaluation of John for over-allocation of work and irresponsibility after anaphora resolution, the overall summary quality does not change much.

We believe that this difference stems from three main factors. First, replacing ambiguous pronouns with explicit noun phrases can greatly reduce the ambiguity of the input and facilitates direct alignment of semantic roles. Second, strengthening the coherence of the text allows the model to learn the correspondence between characters and context more efficiently. For small models, this lightweight preprocessing can significantly reduce the noise during fine-tuning and improve learning effects. Third, the model does not have to remember or learn the antecedents corresponding to different pronouns during training, and perhaps self-attention can be aligned without having to span large distances. However, for models with larger capacity and deep context modeling capabilities, the benefits are relatively diminishing. Moreover, we speculate that pre-training method of destroying the input enables BART to strengthen its understanding of entity and paragraph coherence during the reconstruction process, so the marginal benefit of anaphora resolution is relatively small compared to T5.

7 Conclusions and Future Work

This study investigates whether preprocessing with anaphora resolution before LLM fine-tuning for summary application can improve the model performance. By fine-tuning the T5-base model and the BART-large model on the SAMSum dataset with the original text and the text processed by the simplified version of MARS. The results show that T5-base achieves highly significant gains in ROUGE-1, ROUGE-2, and ROUGE-L metrics after anaphora resolution, which fully demonstrates how anaphora resolution enhances the ability of the model to capture semantic coherence. BART-large, on the other hand, only shows a small and non-significant increase in each metric, indicating that its innate contextual understanding already covers most parsing relationships, and thus has limited marginal benefits.

This study is still limited to the SAMSum dataset and two models. The applicability of other corpora, languages, or larger-scale LLMs remains to be verified. In addition, the interaction between hyperparameters (such as learning rate, number of frozen layers) and the benefits of anaphora resolution also needs to be systematically explored. Future research can further expand to more models and datasets. For example, at the model level, experiments can be conducted using larger LLMs such as GPT-NeoX-20B (Black et al., 2022) or Llama 2 (Touvron et al., 2023). At the data level, different styles and topics of summary datasets such as MeetingBank (Hu et al., 2023) or CNN/DailyMail (Nallapati et al., 2016) can be considered. Furthermore, according to a comparative study by Mitkov and Ha (2024), the use of state-of-the-art anaphora resolution methods based on deep learning (such as DeBERTa-based token labelling) may further improve the accuracy, which in turn may lead to stronger summarisation performance.

Acknowledgments

This work has been partially supported by the CIDEXG/2023/12 project, funded by the Generalitat Valenciana.

References

Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Ivan S Blekanov, Nikita Tarasov, and Svetlana S Bodrunova. 2022. Transformer-based abstractive summarization for reddit and twitter: single posts vs. comment pools in three languages. *Future Internet*, 14(3):69.

Richard Evans. 2001. Applying machine learning toward an automatic classification of it. *Literary and linguistic computing*, 16(1):45–58.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Sixth workshop on very large corpora*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv:2112.07916*.

Ilya Gusev. 2020. Dataset for automatic summarization of russian news. In *Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7–9, 2020, Proceedings 9*, pages 122–134. Springer.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spacy: Industrial-strength natural language processing in python.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. *arXiv preprint arXiv:2305.17529*.

Shalom Lappin and Herbert J Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Ruslan Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Ruslan Mitkov. 2002. *Anaphora resolution*. Routledge.

Ruslan Mitkov. 2022. Anaphora resolution. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, second, substantially revised edition, pages 707–729. Oxford University Press.

Ruslan Mitkov, Richard Evans, and Constantin Orasan. 2002. A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3*, pages 168–186. Springer.

Ruslan Mitkov, Richard Evans, Constantin Orăsan, Iustin Dornescu, and Miguel Rios. 2012. Coreference resolution: To what extent does it help nlp applications? In *Text, Speech and Dialogue: 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings 15*, pages 16–27. Springer.

Ruslan Mitkov, Richard Evans, Constantin Orăsan, Le An Ha, and Viktor Pekar. 2007. Anaphora resolution: To what extent does it help nlp applications? In *Anaphora: Analysis, Algorithms and Applications: 6th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC 2007, Lagos, Portugal, March 29-30, 2007. Selected Papers 6*, pages 179–190. Springer.

Ruslan Mitkov and Le An Ha. 2024. Are rule-based approaches a thing of the past? the case of anaphora resolution. *Procesamiento del Lenguaje Natural*, 73(0):15–27.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Constantin OrĂsan, Richard Evans, and Ruslan Mitkov. 2000. Enhancing preference-based anaphora resolution with genetic algorithms. In *Natural Language Processing—NLP 2000: Second International Conference Patras, Greece, June 2–4, 2000 Proceedings 2*, pages 185–195. Springer.

Chris D Paice and Gareth D Husk. 1987. Towards the automatic recognition of anaphoric features in english text: the impersonal pronoun "it". *Computer Speech & Language*, 2(2):109–132.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

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.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Pasi Tapanainen and Timo Jarvinen. 1997. A non-projective dependency parser. In *Fifth Conference on Applied Natural Language Processing*, pages 64–71.

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*.

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.

Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer.

Hemant Yadav, Nehal Patel, and Dishank Jani. 2023. Fine-tuning bart for abstractive reviews summarization. In *Computational Intelligence: Select Proceedings of InCITe 2022*, pages 375–385. Springer.

Tiezheng Yu, Dan Su, Wenliang Dai, and Pascale Fung. 2020. Dimsum@ laysumm 20: Bart-based approach for scientific document summarization. *arXiv preprint arXiv:2010.09252*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

A Example Dialogue and Model Outputs

Input Dialogue

Igor: Shit, I've got so much to do at work and I'm so demotivated.
John: It's pretty irresponsible to give that much work to someone on their notice period.
Igor: Yeah, exactly! Should I even care?
John: It's up to you, but you know what they say...
Igor: What do you mean?
John: Well, they say how you end things shows how you really are...
Igor: And not how you start, right?
John: Gotcha!
Igor: So what shall I do then?
John: It's only two weeks left, so grit your teeth and do what you have to do.
Igor: Easy to say, hard to perform.
John: Come on, stop thinking, start doing!
Igor: That's so typical of you! ;)

Reference Summary *Igor has a lot of work on his notice period and he feels demotivated. John thinks he should do what he has to do nevertheless.*

A.1 T5-base

Summary from Raw Model *Igor has a lot of work to do. John advises him to stop thinking and start doing.*

Summary from Anaphora-Resolved Model *Igor has a lot of work to do. He is demotivated. John thinks it's irresponsible to give that much work to someone on their notice period.*

A.2 BART-large

Summary from Raw Model *Igor has a lot of work to do at work. John advises him to do what he has to do.*

Summary from Anaphora-Resolved Model *Igor has a lot of work to do at work. John reckons it's irresponsible to give so much work to someone on their notice period.*

Transformers and Large Language Models for Hope Speech Detection: A Multilingual Approach

Diana P. Madera-Espíndola Tecnológico de Monterrey Estado de México, México A01025835@tec.mx

Zoe Caballero-Domínguez Tecnológico de Monterrey Estado de México, México A01747247@tec.mx

Valeria J. Ramírez-Macías Tecnológico de Monterrey Monterrey, México A01636965@tec.mx

Sabur Butt **Hector G. Ceballos**
Tecnológico de Monterrey, Institute for the Future of Education
Monterrey, México Monterrey, México
saburb@tec.mx ceballos@tec.mx

Abstract

With the rise of Generative AI (GenAI) models in recent years, it is necessary to understand how they performed compared with other Deep Learning techniques, across tasks and across different languages. In this study, we benchmark ChatGPT-4 and XML-RoBERTa, a multilingual transformer-based model, as part of the Multilingual Binary and Multiclass Hope Speech Detection within the PolyHope-M 2025 shared task. Furthermore, we explored prompting techniques and data augmentation to determine which approach yields the best performance. In our experiments, XML-RoBERTa frequently outperformed ChatGPT-4. It also attained F1 scores of 0.86 for English, 0.83 for Spanish, 0.86 for German, and 0.94 for Urdu in Task 1, while achieving 0.73 for English, 0.70 for Spanish, 0.69 for German, and 0.60 for Urdu in Task 2.

1 Introduction

Hope speech detection is an emerging area in Natural Language Processing (NLP) that identifies an expectation, desire, or aspiration focused on the future, aimed at a particular or broad event or outcome, which plays a significant role in shaping human behavior, choices, and emotions (Balouchzahi et al., 2023). This task has become increasingly important in the digital age, particularly on social media platforms where content spread can contribute significantly to emotional well-being. Its relevance was especially highlighted during global crises such as the COVID-19 pandemic, in such contexts, fostering a sense of hope through language plays a crucial role in promoting resilience and mental health (Yadav et al., 2023; Surya Sai Eswar et al., 2022).

Recent research efforts have focused on of advanced machine and deep learning techniques to improve the accuracy of hope speech detection (Sidorov et al., 2024; Ahmad et al., 2024). In particular, transformer-based models have been applied to several NLP tasks and have proved a superior performance compared to other state-of-the-art models (Sidorov et al., 2023). However, the widespread adoption of large language models (LLMs) have transformed how text is represented and understood, particularly in multilingual settings (Chakravarthi, 2022; Kadiyala, 2024). This phenomenon has provoked researchers to explore the performance on sentiment analysis of new GenAI models against traditional transformer-based ones (Krugmann and Hartmann, 2024; Anas et al., 2024; Bu et al., 2024).

Despite these efforts, the experiments on the literature explore sentiment analysis broadly and there is no existing research, to the best of our knowledge, comparing GenAI and traditional models in hope classification. Therefore, in this study, we benchmarked ChatGPT-4 against XML-RoBERTa. We chose these specific models due to their popularity and performance in similar studies (Krugmann and Hartmann, 2024; Krasitskii et al., 2024; Shridhara et al., 2023). Furthermore, we explored the effectiveness of various strategies designed to optimize model performance. Specifically, we used one-shot and few-shot prompting techniques on the generative model, and data augmentation for RoBERTa.

Through a detailed evaluation of these approaches, the research provides a comprehensive analysis of how these two models compare to each other when applied to detect hope speech across diverse linguistic settings, including English, Spanish, German, and Urdu, within the framework of the PolyHope-M at RANLP 2025 shared task (Fazlourrahman et al., 2025), which emphasizes the

Our code is publicly available at https://github.com/DianaPME/PolyHope-M-RANLP-2025

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 67–76, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-008

value of harnessing existing multilingual datasets to navigate the complexities of linguistic and cultural diversity in sentiment analysis. Through this approach, it supports efforts to close communication gaps and cultivate safer, more inclusive digital communities.

2 Related Work

Social media platforms play a central role in shaping public discourse and offer a vast repository of user-generated content for linguistic analysis. These platforms provide concise and context-rich data, making them a widely used source for NLP research. Among the popular tasks explored in this domain is hate speech detection, which involves the identification and classification of language that conveys hostility, incites violence, or reinforces harmful stereotypes (Shridhara et al., 2023).

While this task aims to identify and mitigate negative online behavior, another emerging area of research is hope speech detection which serves as a source of encouragement for many people during times of illness, stress, loneliness, or depression (García-Baena et al., 2023; García-Baena et al., 2024), emphasizing the promotion of mental well-being in digital spaces (Zhu, 2022).

Relevant to this emerging task is the growing focus on diversifying the languages represented in hope speech datasets, enabling models to generalize better across linguistic and cultural contexts, support cross-linguistic transfer learning, and capture semantic nuances that vary across cultures. The HopeEDI dataset is one such effort, consisting of English, Malayalam, and Tamil YouTube comments (Chakravarthi, 2020). However, as highlighted by Gowda et al. (2022), creating effective multilingual models for hope speech detection presents substantial challenges, particularly due to language diversity and the presence of various scripts. This underscores the need for techniques such as data augmentation, including back-translation, where text is translated into another language and then back to the original to generate synthetic data. These methods are essential for expanding linguistic coverage and improving model performance in diverse language contexts (LekshmiAmmal et al., 2024).

On this line of research, the IberLEF (García-Baena et al., 2023; García-Baena et al., 2024; Butt et al., 2025) and RANLP (Sidorov et al., 2024; Balouchzahi et al., 2025) workshops on Hope Speech Detection introduce a new multilingual challenge by expanding the understanding of hope speech. It does so through the construction of a corpus that allows for both binary classification, identifying tweets as either Hope or Not Hope, and a more nuanced fine-grained categorization into three distinct types: Generalized Hope, Realistic Hope, and Unrealistic Hope. These efforts make a crucial and challenging contribution by filling a notable gap in annotated datasets dedicated to hope, since existing resources tend to omit it or misclassify it as a generic positive emotion, resulting in inaccurate predictions (Butt et al., 2025). In addition, the task provides a platform to evaluate the capabilities of advanced models in processing data across diverse linguistic contexts (Balouchzahi et al., 2022; Krasitskii et al., 2024).

In automated hope speech detection, various methods have been explored to improve performance. The introduction of transformer-based architectures has significantly impacted advancements in NLP. Models such as BERT, RoBERTa, and DistilBERT have outperformed traditional approaches, as they achieve remarkable results in a variety of applications, including hope speech detection, with the multilingual versions demonstrating the ability to effectively handling a range of languages (Dowlagar and Mamidi, 2021; Hossain et al., 2021; Sidorov et al., 2023).

On the other hand, the increasing use of Generative AI tools, particularly Large Language Models such as GPT 3 and over, has introduced promising possibilities for hope speech detection. These models can be guided using various prompting techniques, including zero-shot prompting, few-shot prompting, and chain-of-thought prompting, to generate relevant and meaningful responses (Thuy and Thin, 2024).

Since the popularization of GenAI, several researchers have been working comparing these models to the more traditional transformer-based models. Krugmann and Hartmann (2024) performed a binary and three-class sentiment classification experiment between GenAI and transformer-based models. Their experiments show that fine-tuned transfer-learning models frequently outperform general-purpose LLMs. Similarly, in an study made by Anas et al. (2024), RoBERTa attained the best performance against GenAI models in product review analysis. However, GenAI have also surpassed transformer-based models in other studies, for example, Konstantinos et al. (2024) concludes that GPT 3.5 is better at product review evaluations than BERT and RoBERTa. In another instance, ChatGPT 3.5 archived the best performance at the IberLEF 2024 hope competition for the binary task, surpassing transformer-based entries (García-Baena et al., 2024).

This experiments showcase that there is still much to learn about the use of Generative AI models for sentiment analysis, not to mention for hope detection or across languages.

3 Dataset

The dataset used in this study is sourced from the PolyHope-M dataset, which is part of the RANLP 2025 shared task `https://www.codabench.org/competitions/5635/`. It extends the original PolyHope dataset (Balouchzahi et al., 2023) by translating its English keywords into Spanish and German, with careful validation by native speakers to ensure linguistic and contextual accuracy. Tweets were collected using the Tweepy API and annotated by three qualified annotators per language, with final labels determined by majority vote. Additionally, also in line with the original PolyHope dataset, (Balouchzahi et al., 2025) replicated its label descriptions and definitions to develop a comparable dataset in Urdu, thereby maintaining consistency with prior work while expanding to a low-resource language. The resulting dataset is a combination of the original English, Spanish, German, and newly created Urdu data, representing the first multiclass hope speech detection dataset covering these four languages. This multilingual dataset enables comprehensive analysis and modeling of hope speech across diverse linguistic and cultural contexts, addressing a critical gap in the literature.

The data provided consists of Twitter texts in English, Spanish, German, and Urdu and is divided into three subsets: a training set, a development set, and a final test set. The development and test datasets each included three columns. One contained the tweet text, another provided the binary classification label (Hope or Not Hope), and the third represented the multiclass classification label (Generalized Hope, Realistic Hope, Unrealistic Hope, or Not Hope). In contrast, the test set included only the tweet text. It is important to note that the distribution across languages was imbalanced, with the number of Spanish and German tweets being approximately twice that of English and Urdu.

4 Methodology

4.1 Data processing

Our first step in the methodology was to clean the data to enhance the performance of the models. The text preprocessing involved standardizing the text to lowercase, trimming extra spaces, eliminating HTTP links, and removing Twitter-specific elements such as user mentions and retweet tags (rt). It also included filtering out non-alphabetical characters specific to each language, deleting emojis that appeared multiple times, and replacing the remaining emojis with their textual descriptions.

4.2 Data augmentation

As previously mentioned, a class imbalance was observed between the languages. To help mitigate this, data augmentation was applied by translating the original Spanish training data into English and the original English training data into Urdu, as only a direct translation pathway from English to Urdu was available. The resulting translated texts were then added to the respective English and Urdu training sets.

For this translation task, we used the Helsinki-NLP pre-trained machine translation model with the MarianMT tokenizer from the HuggingFace library. Specifically, we used *"Helsinki-NLP/opus-mt-en-ur"* for English to Urdu and *"Helsinki-NLP/opus-mt-es-en"* for Spanish to English. The translation was executed on a Google Colab environment with GPU support, using the free-tier account. This model was selected for its ease of implementation and efficient inference times.

4.3 XLM-Roberta

For the XLM-RoBERTa model, we converted the labels into numerical values and used a merged training set that combined all four languages. The training parameters used were: number of train epochs: 3, learning rate: 1e-5, and max sequence length: 64. These parameters were selected through trial and error, given the limited computational resources available. We utilized Google Colab with a GPU, but due to constraints on the number of available GPU units, we were limited by the parameters allowed in this configuration. Nevertheless, the parameters were primarily based on those used in the study presented by (Qu et al., 2021).

4.4 ChatGPT-4

We used the GPT-4 model that was available since 2023 but was discontinued on April 2025. Due to the limited number of available tokens, we chose to use the UI or chat versions instead. For each subtask and language, a specific prompt was defined, which will be explained in the next subsection. Furthermore, taking advantage of the model's chat capabilities, we provided the dataset in batches for classification.

4.4.1 Zero-Shot Prompts

For the zero-shot prompts, we adopted a unified approach, using the same prompt for binary classification in the four languages. Similarly, a single prompt was designed for the multiclass classification task across all languages. This decision was made under the assumption that the model would generalize the task regardless of the input language. To further assist the model, we included the class descriptions provided on the contest page directly within the prompt for clearer guidance. The prompts used are shown below.

Binary Classification Prompt

Below, there is a list of lines of text. Your job is to decide whether the given text reflects hope or lack of hope by classifying it as either Hope or Not Hope.
The definitions are:
-Hope: Hope is a crucial human emotion that influences decision-making, resilience, and social interactions.
-Not Hope: Not Hope is a text that does not express hope.
Please, give the answer in the format ”number, classification”. Don't forget the comma instead of a dot in your answer
Text to classify

Multiclass Classification Prompt

Below, there is a list of lines of text.Your job is to classify the text as a Generalized Hope, Realistic Hope, Unrealistic Hope, or Not Hope. The definitions are:
- Generalized Hope: A broad sense of optimism not tied to specific outcomes.
- Realistic Hope: Expectations grounded in achievable goals. - Unrealistic Hope: Desires for outcomes that are unlikely or impossible.
- Not Hope: Not hope is that belonged to neither category above. Texts that do not express hope.
Please, give the answer in the format ”number, classification”. Don't forget the comma instead of a dot in your answer
Text to classify

4.4.2 Few-Shot Prompts

For the few-shot prompts, we selected three random samples from each class in the training set, creating three example shots. We opted to use separate prompts for each language, as the examples would be specific to each language. The structure of the prompt is consistent with the zero-shot classification; the only difference is the inclusion of examples with both text and labels, which vary depending on the language. The same set of examples was used across all models.

5 Results

We evaluated the performance of the fine-tuned XML-RoBERTa model against ChatGPT-4 in hope speech detection in four languages: English, Spanish, German, and Urdu. This evaluation was performed over two tasks: binary and multiclass. The binary task was measured using accuracy and macro-averaged F1-score as evaluation metrics, while the multiclass task used accuracy and weighted-average F1-score.

XML-RoBERTa consistently outperformed ChatGPT-4 across languages and tasks (Figure 1 and Figure 2). Table 1 shows that RoBERTa without data augmentation achieved the highest performance in both tasks in the English set. Similarly, in Spanish (Table 2, RoBERTa without augmentation again led binary classification. But the pattern breaks in multiclass classification, where RoBERTa yielded the best F1-score, while RoBERTa trained with data augmentation obtained the highest accuracy.

For the German and Urdu datasets, RoBERTa also outperformed ChatGPT-4. In German, the data augmentation version had better performance in multiclass task, while the single version in binary (Table 3). In the case of Urdu, as shown in Table

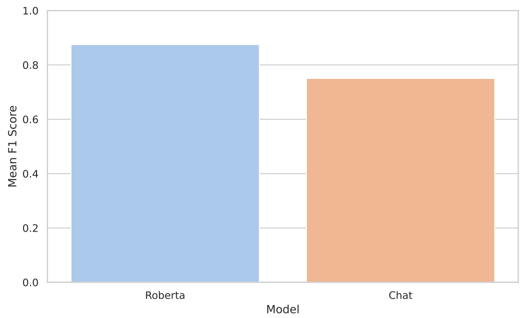


Figure 1: Average F1-score for both models, ChatGPT-4 (2023-2025) and RoBERTa, in the binary hope detection task.

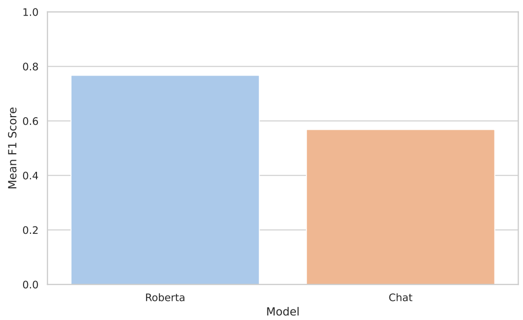


Figure 2: Average F1-score for both models, ChatGPT-4 and RoBERTa, in the multiclass hope detection task.

4) the single version slightly outperformed its augmented counterpart in both tasks. Notably, all performance differences between RoBERTa models with and without data augmentation were minimal.

Figures 3 and 4 clearly show that both models performed better on the binary classification task compared to the multiclass one. However, there is no clear trend regarding the effectiveness of zero-shot versus few-shot prompting strategies for ChatGPT-4. Although few-shot prompting yielded slightly better results in 5 out of 8 evaluations, the differences were not substantial.

Regarding the effect of data augmentation, the results suggest that RoBERTa's performance remains largely stable regardless of its inclusion. A slight advantage was noted for the non-augmented model across tasks.

For the test set predictions, we submitted RoBERTa results with and without data augmentation. Augmentation improved Spanish results, hurt Urdu performance, and had negligible impact on English and German. Tables 5, 6, 7, and 8 present a comparison between the top five places in the competition. Our scores secured a place on the leaderboard for all tasks across the four languages.

(a) Binary Task

Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.6851	0.6860
Model 1	Few	0.7338	0.7351
Model 2	NA	**0.8433**	**0.8436**
Model 2	Aug.	0.8415	0.8418

(b) Multiclass Task

Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.5428	0.5721
Model 1	Few	0.5453	0.5648
Model 2	NA	**0.7497**	**0.7460**
Model 2	Aug.	0.745261	0.7418

Table 1: Results on the English dataset across all models and combinations of tasks and strategies. ChatGPT-4 is denoted as Model 1; XLM-Roberta as Model 2. "Aug" refers to data augmentation applied to the training set.

(a) Binary Task

Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.7006	0.7010
Model 1	Few	0.6949	0.6958
Model 2	NA	**0.8432**	**0.8433**
Model 2	Aug	0.8405	0.8407

(b) Multiclass Task

Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.5095	0.4873
Model 1	Few	0.5439	0.5097
Model 2	NA	**0.7572**	0.7529
Model 2	Aug	0.7533	**0.7479**

Table 2: Results on the Spanish dataset across all models and combinations of tasks and strategies. ChatGPT-4 is denoted as Model 1; XLM-Roberta as Model 2. "Aug" refers to data augmentation applied to the training set.

Specifically, we achieved first place in both binary and multiclass tasks for English, fifth and first place for Spanish binary and multiclass tasks respectively, fourth and second place for German, and sixth and fourth place for Urdu binary and multiclass tasks respectively.

Figure 5 shows the confusion matrices obtained on the development set for RoBERTa across languages. In English, the model more accurately predicts Hope than Not Hope, but tends to misclassify Hope as Not Hope more often, reflecting a slight

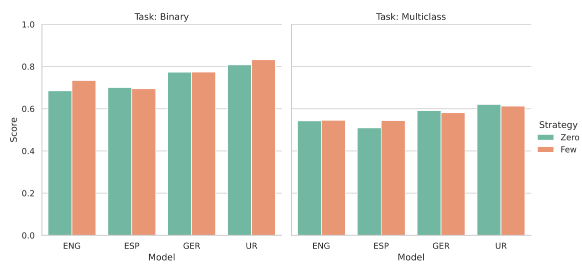


Figure 3: F1 scores for zero vs few prompt strategy with ChatGPT-4 (2023-2025) across all datasets.

(a) Binary Task			
Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.7734	0.7773
Model 1	Few	0.7740	0.7785
Model 2	NA	**0.8813**	**0.8830**
Model 2	Aug	0.8778	0.8797
(b) Multiclass Task			
Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.5912	0.5682
Model 1	Few	0.5814	0.5546
Model 2	NA	0.8344	0.833888
Model 2	Aug	**0.8350**	**0.8343**

Table 3: Results on the German dataset across all models and combinations of tasks and strategies. ChatGPT-4 is denoted as Model 1; XLM-Roberta as Model 2. "Aug" refers to data augmentation applied to the training set.

(a) Binary Task			
Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.8082	0.8110
Model 1	Few	0.8323	0.8343
Model 2	NA	**0.9456**	**0.9457**
Model 2	Aug	0.9268	0.9272
(b) Multiclass Task			
Model	**Strategy**	**F1-Score**	**Accuracy**
Model 1	Zero	0.6203	0.6227
Model 1	Few	0.6127	0.6239
Model 2	NA	**0.7547**	**0.7586**
Model 2	Aug	0.7085	0.7109

Table 4: Results on the Urdu dataset across all models and combinations of tasks and strategies. ChatGPT-4 is denoted as Model 1; XLM-Roberta as Model 2. "Aug" refers to data augmentation applied to the training set.

bias toward Hope. In the Spanish and German sets, while in Spanish RoBERTa is better at classifying Hope contrarily to German, the model makes more frequent errors misclassifying Hope than Not Hope. And Urdu shows the most balanced results.

Analyzing the multiclass confusion matrices in Figure 6, we observe distinct performance patterns across languages. In English, the model most accurately classifies Not Hope, followed by moderate success with Generalized Hope, and lower accuracy for Realistic Hope and Unrealistic Hope.

Figure 4: F1 scores for normal and data augmentation-trained RoBERTa across all datasets.

(a) Binary Task		
User name	**Acc**	**Avg Mac F_1**
dmadera	**0.8634**	**0.8632**
nomanjaffar11	0.8629	0.8629
oluwatobi	0.8610	0.8608
julkarnaeen	0.8610	0.8606
teddymas	0.8557	0.8548
(b) Multiclass Task		
User name	**Acc**	**Avg Mac F_1**
dmadera	**0.7801**	**0.7304**
nomanjaffar11	0.7729	0.7121
priya27	0.7680	0.7111
ahmedembedded	0.7622	0.7028
teddymas	0.7457	0.6999

Table 5: Comparison with top 5 results in the competition for Task 1 and Task 2 for English.

(a) Binary Task		
User name	**Acc**	**Avg Mac F_1**
nomanjaffar11	0.8499	0.8498
teddymas	0.8479	0.8478
julkarnaeen	0.8407	0.8407
priiyo9	0.8405	0.8404
dmadera	**0.8334**	**0.8326**
(b) Multiclass Task		
User name	**Acc**	**Avg Mac F_1**
dmadera	**0.7660**	**0.7067**
teddymas	0.7358	0.6856
nomanjaffar11	0.7533	0.6856
abit7431	0.7377	0.6711
priiyo9	0.7433	0.6706

Table 6: Comparison with top 5 results in the competition for Task 1 and Task 2 for Spanish.

The most frequent confusions involve Generalized Hope being misclassified as Realistic Hope and as Not Hope. For Spanish, the model performs strongly on Not Hope and reasonably well on Generalized Hope, but struggles more with Realistic Hope and Unrealistic Hope. On the other hand, for German, the model excels at identifying both Not Hope and Generalized Hope, while achieving moderate accuracy on Realistic Hope and performing poorly on Unrealistic Hope. Finally, for Urdu, the model shows strong performance on Not Hope, decent accuracy on Generalized Hope and Unrealistic Hope, but severely underperforms on Realistic Hope. The most frequent misclassifications are between Generalized Hope and Unrealistic Hope

6 Discussion

The results indicate that RoBERTa consistently outperformed ChatGPT-4 across most tasks and languages, particularly in the more structured binary classification setting. These findings are consistent with prior research in related NLP tasks, where fine-tuned supervised transformers such as RoBERTa

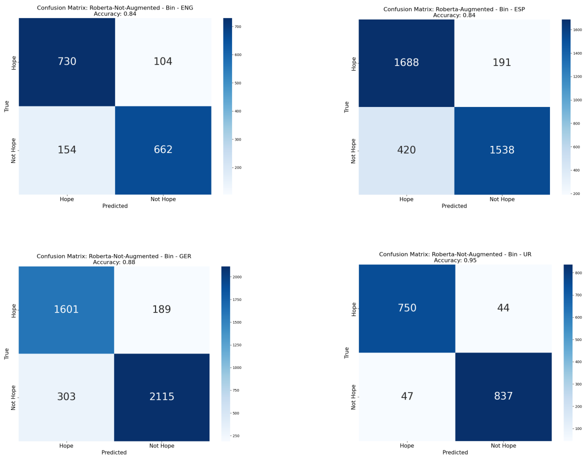


Figure 5: Confusion matrices for the binary classification task of the best model (XLM-RoBERTa) across four languages: English (top left), Spanish (top right), German (bottom left), and Urdu (bottom right).

Figure 6: Confusion matrices for the multiclass classification task of the best model (XLM-RoBERTa) across four languages: English (top left), Spanish (top right), German (bottom left), and Urdu (bottom right).

(a) Binary Task

User name	**Acc**	**Avg Mac F_1**
teddymas	0.8746	0.8726
nomanjaffar11	0.8742	0.8715
abit7431	0.8668	0.8638
dmadera	**0.8647**	**0.8633**
unstoppable	0.8576	0.8568

(b) Multiclass Task

User name	**Acc**	**Avg Mac F_1**
nomanjaffar11	0.8345	0.7013
dmadera	**0.8229**	**0.6968**
teddymas	0.8135	0.6944
abit7431	0.8172	0.6778
julkarnaeen	0.8004	0.6741

Table 7: Comparison with top 5 results in the competition for Task 1 and Task 2 for German.

(a) Binary Task

User name	**Acc**	**Avg Mac F_1**
abit7431	0.9499	0.9498
nomanjaffar11	0.9499	0.9498
teddymas	0.9499	0.9498
oluwatobi	0.9480	0.9480
ahmedembedded	0.9461	0.9461
dmadera	**0.9451**	**0.9451**

(b) Multiclass Task

User name	**Acc**	**Avg Mac F_1**
nomanjaffar11	0.7836	0.6526
abit7431	0.7736	0.6482
teddymas	0.7655	0.6314
dmadera	**0.7769**	**0.6079**
priiyo9	0.7636	0.6015

Table 8: Comparison with top 5 results in the competition for Task 1 and Task 2 for Urdu.

often outperform Large Language Models (LLMs) on tasks requiring nuanced understanding of short texts. For instance, Krugmann et al. (2024) and Zhang (2024) highlight that models like SiEBERT and RoBERTa excel on short-form content where LLMs tend to struggle.

Furthermore, our results are also similar to those in the baseline experiments made by Sidorov et al. (2024) and Balouchzahi et al. (2025). Table 9 shows the comparison between the baseline results obtained with RoBERTa variants presented by the authors and our Test set results. Sidorov et al. used the model *"FacebookAI/xlm-roberta-base"* for English, Spanish, and German. Balouchzahi et al. used *"urduhack/roberta-urdu-small"* for the Urdu dataset. All models were obtained through HuggingFace. Meanwhile, we used the same model for all datasets (*"xlm-roberta-large"*) and fine-tuned it using the training set plus data augmentation. As the table shows, we outperformed the baseline results, confirming that XML-RoBERTa is a powerful transformer for hope detection and that the

(a) Binary Task

Authors	ENG	ES	GER	UR
1	0.8623	0.8369	0.8704	-
2	-	-	-	0.6961
3	0.8632	0.8326	0.8633	0.9451

(b) Multiclass Task

Authors	ENG	ES	GER	UR
1	0.6907	0.6801	0.6878	-
2	-	-	-	0.4801
3	0.7304	0.7067	0.6968	0.6079

Table 9: Comparison of Avg Macro F_1 scores between baseline results from Sidorov et al. (1), and Balouchzahi et al. (2), and our proposed method (Madera et al. (3)) evaluated on the test set.

addition of data augmentation can lead to better performance.

Interestingly, our experiments showed a strong performance in under-resourced languages such as Urdu and German, an unexpected outcome given that English and Spanish are more prominently represented in large-scale datasets and benchmarks (Balouchzahi et al., 2025). The binary confusion matrices indicate that linguistic features or language-specific training data characteristics influence how the model allocates predictions between the two classes, with German and Spanish showing the strongest biases toward 'Not Hope' compared to English and Urdu. The fact that both models maintained reasonable effectiveness across these languages suggests that multilingual models like XLM-RoBERTa can successfully transfer knowledge to underrepresented languages. However, further investigation is needed to confirm this trend and to ensure equitable performance across diverse linguistic contexts.

In contrast, for the multiclass classification task, all models perform best at predicting 'Not Hope', with the exception of German, where the model excels. Across Spanish, German, and Urdu, 'Realistic Hope' consistently emerges as the most challenging class to predict. This multiclass analysis highlights the model's difficulties in distinguishing nuanced hope categories across languages, with each language exhibiting distinct patterns of confusion between specific class pairs.

It is important to note that we trained RoBERTa with scarce computational resources and a short time-period. Therefore, while we obtained superior results, these can be improve with prolonged training or further hyperparameter optimization.

Finally, regarding ChatGPT-4, we recommend exploring additional prompting techniques and test in smaller batch settings. Generative AI has great potential for sentiment analysis and its continuous growth in use (Kim, 2024), including cases of emotional companionship, justifies the need for continued research on how they can detect complex emotions such as hope.

7 Conclusion

Hope speech detection is a growing field in NLP that seeks to identify expressions of expectation, aspiration, or encouragement. These emotions have a key role on human behavior and emotional well-being (Balouchzahi et al., 2023). In the present study, we evaluate and compare the effectiveness of two approaches: transformer-based model XML-RoBERTa with and without data augmentation, and the generative large language model ChatGPT-4 using zero-shot and few-shot prompting. We use the multilingual dataset provided by the PolyHope-M shared task at RANLP 2025, and assess both binary and multiclass classification tasks across English, Spanish, German, and Urdu.

The results demonstrate that RoBERTa consistently outperformed ChatGPT-4 across all tasks and languages, with notable higher performance in the binary classification setting. These findings support prior evidence that supervised model remain highly effective for short-text emotion detection, while LLMs may struggle due to their context dependence. For future work, we suggest exploring other prompts that leverage the LLMs generative abilities for better classification, as well as further hyperparameter optimization for RoBERTa models.

References

M.A. Ahmad, Sardar Usman, Farid Humaira, Ameer Iqra, Muhammad Muzzamil, Ameer Hmaza, Grigori Sidorov, and Ildar Batyrshin. 2024. Hope speech detection using social media discourse (posi-vox-2024): A transfer learning approach. *Journal of language and education*, 10:31 – 43.

Mohammad Anas, Anam Saiyeda, Shahab Saquib Sohail, Erik Cambria, and Amir Hussain. 2024. Can generative ai models extract deeper sentiments as compared to traditional deep learning algorithms? *IEEE Intelligent Systems*, 39(2):5–10.

Fazlourrahman Balouchzahi, Sabur Butt, Maaz Amjad, Grigori Sidorov, and Alexander Gelbukh. 2025. Urduhope: Analysis of hope and hopelessness in urdu texts. *Knowledge-Based Systems*, 308:112746.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2022. Polyhope: Two-level hope speech detection from tweets.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2023. Polyhope: Two-level hope speech detection from tweets. *Expert Systems with Applications*, 225:120078.

Kun Bu, Yuanchao Liu, and Xiaolong Ju. 2024. Efficient utilization of pre-trained models: A review of sentiment analysis via prompt learning. *Knowledge-Based Systems*, 283:111148.

Sabur Butt, Fazlourrahman Balouchzahi, Ahmad Imam Amjad, Maaz Amjad, Hector G. Ceballos, and Salud Maria Jimenez-Zafra. 2025. Optimism, expectation, or sarcasm? multi-class hope speech detection in spanish and english.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2022. Multilingual hope speech detection in english and dravidian languages. *International Journal of Data Science and Analytics*, 14(4):389–406.

Suman Dowlagar and Radhika Mamidi. 2021. EDIOne@LT-EDI-EACL2021: Pre-trained transformers with convolutional neural networks for hope speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 86–91, Kyiv. Association for Computational Linguistics.

Balouchzahi Fazlourrahman, Sabur Butt, Maaz Amjad, Luis Jose Gonzalez-Gomez, Abdul Gafar Manuel Meque, Helena Gomez-Adorno, Bharathi Raja Chakravarthi, Grigori Sidorov, Thomas Mandl, Ruba Priyadharshini, Hector Ceballos, and Saranya Rajiakodi. 2025. Overview of PolyHope-M at RANLP: Bridging Hope Speech Detection Across Multiple Languages. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing*.

Daniel García-Baena, Fazlourrahman Balouchzahi, Sabur Butt, Miguel Ángel García Cumbreras, Atnafu Lambebo Tonja, José Antonio García-Díaz, Selen Bozkurt, Bharathi Raja Chakravarthi, Hector G. Ceballos, Rafael Valencia-García, Grigori Sidorov, Luis Alfonso Ureña López, Alexander F. Gelbukh, and Salud María Jiménez-Zafra. 2024. Overview of hope at iberlef 2024: Approaching hope speech detection in social media from two perspectives, for equality, diversity and inclusion and as expectations. *Proces. del Leng. Natural*, 73:407–419.

Daniel García-Baena, Miguel Ángel García-Cumbreras, Salud María Jiménez-Zafra, José Antonio García-Díaz, and Rafael Valencia-García. 2023. Hope speech detection in spanish: The lgbt case. *Language Resources and Evaluation*, 1:1–28.

Anusha Gowda, Fazlourrahman Balouchzahi, H. L. Shashirekha, and Grigori Sidorov. 2022. Mucic@lt-edi-acl2022: Hope speech detection using data re-sampling and 1d conv-lstm. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 161 – 166. Association for Computational Linguistics.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2021. Nlp-cuet@lt-edi-eacl2021: Multilingual code-mixed hope speech detection using cross-lingual representation learner. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 168 – 174. Association for Computational Linguistics.

Ram Mohan Rao Kadiyala. 2024. Cross-lingual emotion detection through large language models. In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 464–469, Bangkok, Thailand. Association for Computational Linguistics.

Minseong Kim. 2024. Unveiling the e-servicescape of chatgpt: Exploring user psychology and engagement in ai-powered chatbot experiences. *Behavioral Sciences*, 14:558.

Mikhail Krasitskii, Olga Kolesnikova, Liliana Hernandez, Grigori Sidorov, and Alexander Gelbukh. 2024. Hope2024@iberlef: A cross-linguistic exploration of hope speech detection in social media. In *40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024)*, volume 459, pages 407–419.

Jan Ole Krugmann and Jochen Hartmann. 2024. Sentiment analysis in the age of generative ai. *Customer Needs and Solutions*, 11(1):3.

Hariharan RamakrishnaIyer LekshmiAmmal, Manikandan Ravikiran, Gayathri Nisha, Navyasree Balamuralidhar, Adithya Madhusoodanan, Anand Kumar Madasamy, and Bharathi Raja Chakravarthi. 2024. Overlapping word removal is all you need: revisiting data imbalance in hope speech detection. *Journal of Experimental & Theoretical Artificial Intelligence*, 36(8):1837–1859.

Yuanchi Qu, Yanhua Yang, and Gang Wang. 2021. Ynu qyc at meoffendes@iberlef 2021: The xlm-roberta and lstm for identifying offensive tweets. In *IberLEF@SEPLN*.

Konstantinos I. Roumeliotis, Nikolaos D. Tselikas, and Dimitrios K. Nasiopoulos. 2024. Llms in e-commerce: A comparative analysis of gpt and llama models in product review evaluation. *Natural Language Processing Journal*, 6:100056.

Manohar Gowdru Shridhara, Viktor Pristaš, Albert Kotvytskiy, L'ubomír Antoni, and Gabriel Semanišin. 2023. A short review on hate speech detection: challenges towards datasets and techniques. In *2023 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 204–209.

Grigori Sidorov, Fazlourrahman Balouchzahi, Sabur Butt, and Alexander Gelbukh. 2023. Regret and hope on transformers: An analysis of transformers on regret and hope speech detection datasets. *Applied Sciences*, 13(6).

Grigori Sidorov, Fazlourrahman Balouchzahi, Luís António de Oliveira Ramos, Helena Gómez-Adorno, and Alexander Gelbukh. 2024. Mind-hope: Multilingual identification of nuanced dimensions of hope. *Research Square*.

Medicharla Dinesh Surya Sai Eswar, Nandhini Balaji, Vedula Sudhanva Sarma, Yarlagadda Chamanth Krishna, and Thara S. 2022. Hope speech detection in tamil and english language. In *2022 International Conference on Inventive Computation Technologies (ICICT)*, pages 51–56.

Nguyen Thi Thuy and Dang Van Thin. 2024. An empirical study of prompt engineering with large language models for hope detection in english and spanish. In *IberLEF@SEPLN*.

Neemesh Yadav, Mohammad Aflah Khan, Diksha Sethi, and Raghav Sahni. 2023. Beyond negativity: Re-analysis and follow-up experiments on hope speech detection.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. Sarcasmbench: Towards evaluating large language models on sarcasm understanding.

Yue Zhu. 2022. LPS@LT-EDI-ACL2022:an ensemble approach about hope speech detection. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 183–189, Dublin, Ireland. Association for Computational Linguistics.

Beyond BLEU: Ethical Risks of Misleading Evaluation in Domain-Specific QA with LLMs

Ayoub Nainia[1] **Régine Vignes-Lebbe**[1] **Hajar Mousannif**[2] **Jihad Zahir**[2,3]
[1]Institut de Systématique, Evolution, Biodiversité (ISYEB), Sorbonne Université, Muséum National d'Histoire Naturelle, CNRS, EPHE-PSL, Université des Antilles, F-75005, Paris, France
[2]LISI Laboratory, Cadi Ayyad University, Marrakesh, Morocco
[3]UMMISCO, IRD, France
ayoub.nainia@sorbonne-universite.fr,
regine.vignes_lebbe@sorbonne-universite.fr,
mousannif@uca.ac.ma, j.zahir@uca.ac.ma

Abstract

Large Language Models (LLMs) are increasingly used in scientific question answering (QA), including high-stakes fields such as biodiversity informatics. However, standard evaluation metrics such as BLEU, ROUGE, Exact Match (EM), and BERTScore remain poorly aligned with the factual and domain-specific requirements of these tasks. In this work, we investigate the gap between automatic metrics and expert judgment in botanical QA by comparing metric scores with human ratings across five dimensions: accuracy, completeness, relevance, fluency, and terminology usage. Our results show that standard metrics often misrepresent response quality, particularly in the presence of paraphrasing, omission, or domain-specific language. Through both quantitative analysis and qualitative examples, we show that high-scoring responses may still exhibit critical factual errors or omissions. These findings highlight the need for domain-aware evaluation frameworks that incorporate expert feedback and raise important ethical concerns about the deployment of LLMs in scientific contexts.

1 Introduction

Large language models (LLMs) are increasingly fine-tuned and deployed for question answering (QA) in specialized domains such as biodiversity, medicine, and scientific research. These models offer compelling fluency and broad generalization capabilities, making them attractive for automating knowledge access in fields where information is complex and rich in terminology. However, evaluating their effectiveness in the real-world, especially in high-stakes contexts, remains a critical challenge.

Despite impressive reported performance, most QA systems are evaluated using lexical overlap metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), or Exact Match (EM) (Rajpurkar et al., 2016a). These metrics, while easy to compute, have well-documented limitations: they reward surface similarity over factual accuracy, fail to penalize hallucinated content, and systematically favor longer redundant answers that may appear plausible but lack precision (An et al. (2024b); Maynez et al. (2020)). In scientific and technical domains where answers must be both correct and complete, such metrics can inflate perceived performance and mask serious factual deficiencies.

This issue is especially pronounced in domain-specific Question Answering (QA), where small inaccuracies, such as an incorrect botanical trait or a misrepresented medical guideline, can undermine the reliability of the entire system. Recent studies in medical QA (Singhal et al. (2023); Moor et al. (2023)) and scientific QA (Taylor et al., 2022a) demonstrate that even fine-tuned LLMs often generate answers that sound correct but are either partially wrong, incomplete, or not grounded in verifiable sources. However, these limitations are rarely visible in standard evaluation scores, leading to misguided claims about model readiness and potential misuse in real-world deployments.

In this paper, we critically examine how current evaluation practices contribute to an overestimation of fine-tuned LLM performance in domain-specific QA tasks. Our analysis focuses on botanical trait extraction, a high-stakes scientific application where factual precision and accurate use of terminology are essential. However, the evaluation challenges we highlight are not limited to botany. They also apply to fields such as medicine and law, where even small factual errors can have serious consequences (Singhal et al., 2022; Weidinger et al., 2021). In legal contexts, for example, recent efforts have emphasized the importance of expert-annotated datasets and domain-tuned models to ensure accurate interpretation of statutes and

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 77–86, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-009

regulations (Al Mouatamid et al., 2023).

Biodiversity data, for example, serves as the foundation for ecological research, conservation policy, endangered species monitoring, and climate impact studies. Errors in trait extraction can propagate into global biodiversity databases such as the Global Biodiversity Information Facility (GBIF)[1], leading to misclassifications, flawed scientific conclusions or misinformed policy decisions. Even minor hallucinations or omissions (e.g., in leaf morphology or species distribution) can distort downstream analysis or fieldwork.

We analyze cases where model outputs receive high automatic scores but fail expert evaluation due to factual inaccuracies, incompleteness, or loss of critical context. We propose a set of evaluation principles for scientific QA that prioritize factual faithfulness, information coverage, and grounding in verifiable sources, dimensions often invisible to surface-level metrics like BLEU or EM.

Our findings highlight the need to move beyond BLEU and toward evaluation frameworks that reflect the true utility and limits of LLMs in high-precision domains.

2 Background and Motivation

Automated question answering (QA) systems, including fine-tuned large language models (LLMs), are commonly evaluated using lexical overlap metrics originally developed for tasks such as machine translation and summarization. Among the most widely adopted are BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and Exact Match (EM) (Rajpurkar et al., 2016a) and token-level F1 from extractive QA benchmarks such as SQuAD (Rajpurkar et al., 2016a). These metrics compute n-gram overlap between system outputs and reference answers, either as precision (BLEU), recall (ROUGE), or strict equality (EM). Their popularity is largely due to their ease of implementation, reproducibility, and long-standing use in benchmark comparisons.

However, a growing body of work has questioned the adequacy of these metrics in generative QA settings, where answers are open-ended, multi-sentence, and potentially phrased in ways not captured by reference strings. BLEU and ROUGE focus on surface-level n-gram similarity and do not assess whether an answer is factually correct, complete, or grounded. For example, Maynez et al. (2020) showed that summarization models frequently hallucinate content that is not supported by the source text, but still receive high ROUGE scores. An et al. (2024b) found similar trends in long-form QA: answers that are fluent but incorrect or incomplete are often rewarded by BLEU and ROUGE, while semantically valid but lexically diverse answers are penalized. These findings are echoed in previous critiques (Yang et al., 2018) that demonstrated that metrics such as BLEU and ROUGE poorly capture answer quality in both yes/no and entity-centric QA formats.

Despite these limitations, overlap-based metrics remain dominant, including in domain-specific QA systems. Biomedical, legal, and scientific QA models routinely report BLEU, ROUGE, and Exact Match (EM) as primary evaluation metrics (Lee et al., 2021) (Singhal et al., 2023), often without rigorous human evaluation or claim-level verification (Thorne et al., 2018). In practice, this can lead to inflated perceptions of model performance, especially when answers contain hallucinated or missing information that metrics fail to penalize. This risk is amplified in high-stakes domains such as medicine or biodiversity science, where users may trust a model's fluent output without realizing that it lacks factual correctness or critical details.

The continued reliance on these metrics presents not only a technical concern, but an ethical one Ferdaus et al. (2024). By overstating model reliability, current evaluation practices may contribute to misleading claims of safety and readiness, potentially enabling misuse or over-deployment in sensitive contexts. As LLMs are increasingly proposed as tools for scientific assistance and clinical support, evaluating them using metrics that do not reflect truthfulness, completeness, or verifiability is insufficient and potentially dangerous.

3 Related Work

As large language models (LLMs) are increasingly deployed in high-stakes domains, their evaluation has become a focal point of methodological concern and ethical debate. This section reviews work on QA evaluation metrics, factuality assessment, domain-specific QA challenges, and the responsible deployment of LLMs. Our contribution builds on these foundations by examining how inadequate metrics can systematically misrepresent the real capabilities of fine-tuned models in scientific contexts.

[1] https://www.gbif.org

3.1 Evaluation Metrics for LLM Question Answering

Traditional QA evaluation is heavily based on n-gram overlap metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and Exact Match (EM) (Rajpurkar et al., 2016b). However, these metrics often fail to capture semantic correctness, factual consistency, or completeness, especially in open-ended or domain-specific QA tasks. Studies have shown that token-level overlap correlates poorly with human judgment on complex QA (Yang et al., 2018; Maynez et al., 2020)). The L-Eval benchmark introduced by An et al. (2024a) further demonstrated that BLEU and ROUGE do not align with human preferences, particularly in long-context reasoning tasks. They are also biased toward verbose, lexically similar outputs, further inflating scores for answers that may be inaccurate or incomplete.

More sophisticated semantic similarity metrics, such as BERTScore (Zhang et al., 2020), address paraphrastic variation, but remain sensitive to domain-specific terminology and formatting. In response, some QA evaluations now combine overlap-based metrics with embedding-based similarity and human assessment. There is also increasing interest in using LLMs themselves as evaluators (LLM-as-a-judge) (Zheng et al., 2023), although these introduce new biases (Dubois et al., 2025). In general, there is a growing consensus that surface-level metrics are insufficient to capture factual accuracy in generative QA.

3.2 Factuality and Hallucination Detection in LLMs

In light of these limitations, recent research has focused on factuality: whether generated answers are supported by verifiable evidence. Maynez et al. (2020) and Pagnoni et al. (2021) found that summarization systems often hallucinate content while scoring highly on ROUGE. These findings motivated the development of claim-level verification benchmarks such as FRANK (Pagnoni et al., 2021), HaDeS (HAllucination DEtection dataSet) (Liu et al., 2022), and TruthfulQA (Lin et al., 2022), which assess hallucination at the token level.

Several methods now use retrieval-augmented QA (Lewis et al., 2021), natural language inference, or question decomposition to verify generated content (e.g., QAFactEval (Fabbri et al., 2022), RefChecker (Hu et al., 2024), Attributable to Identified Sources (AIS) (Rashkin et al., 2023)). However, even retrieval-based systems can hallucinate when the retrieved content is incomplete or ambiguous (Moor et al., 2023). Hence, detection remains challenging, with token-level hallucination detectors achieving only ∼70% F1 in specialized domains, indicating that hallucination remains a persistent issue even with dedicated detectors.

3.3 Evaluation in Domain-Specific QA Systems

Evaluating QA systems, in scientific domains, introduces unique challenges. Domain-specific LLMs such as Microsoft's BioGPT (Luo et al., 2022) and Meta's Galactica (Taylor et al., 2022b) perform well on tailored benchmarks but require expert-informed evaluation to ensure factual grounding (Singhal et al., 2023; Bélisle-Pipon, 2024). In medicine, for example, Med-PaLM's evaluation combined human review with metrics to assess not just correctness, but also reasoning quality, potential harm, and trustworthiness (Singhal et al., 2022). However, hallucinations and omissions persisted, where LLMs struggled to contextualize general knowledge into actionable recommendations.

In botany and biodiversity informatics, research is emerging on LLM-based extraction of scientific facts from unstructured text, with recent studies achieving over 90% precision in tasks such as species identification, geocoding, and data structure (Castro et al., 2024). However, these results often mask persistent challenges that standard metrics fail to capture. Current LLMs show a concerning tendency for delivering incorrect information that raises concerns about their reliability in ecological research applications (Gougherty and Clipp, 2024). At the same time, extracting structured knowledge from scientific text remains fundamentally challenging even for fine-tuned models (Dagdelen et al., 2024).

The field faces several domain-specific obstacles that standard metrics do not address. Term ambiguity represents a major challenge, as ecological and botanical terminology often carries context-dependent meanings that LLMs struggle to disambiguate correctly. Domain-specific syntax further complicates extraction, as scientific literature employs specialized linguistic patterns and taxonomic conventions that differ markedly from general text. Additionally, the propagation of subtle errors poses particular risks in scientific contexts, where small inaccuracies can compound into significant misrepresentations of ecological relationships or species characteristics.

Perhaps most critically, current benchmarks often fall short in capturing the diverse behavior of these models in real-world applications, with existing frameworks being limited by their focus on general-purpose queries and lack of diversity across specialized domains (Raju et al., 2024). The absence of curated benchmarks specifically designed for biodiversity informatics, combined with limited human-in-the-loop evaluation frameworks, makes it difficult to reliably assess model factuality, completeness, or risk of systematic errors in scientific knowledge extraction. Although domain-specific datasets such as FloraNER have emerged for botanical named entity recognition (Nainia et al., 2024), these represent only narrow aspects of the broader challenge of biodiversity informatics, leaving significant evaluation gaps in other critical areas such as ecological relationship extraction, species behavior analysis, and cross-domain knowledge integration. This evaluation gap is particularly concerning given that scientific problem-solving requires domain expertise, understanding of long-context information, and multi-step reasoning (Cui et al., 2025) that may not be adequately tested by existing metrics (Dorm et al., 2025).

3.4 Responsible Use and Deployment in High-Stakes Domains

Ethical concerns about LLM deployment have intensified in law, science, and medicine, where overreliance on fluent but inaccurate outputs has led to misinformation, bogus citations, and incorrect legal filings (Weidinger et al., 2021). Therefore, Scholars have called for stricter evaluation, transparency, and oversight, especially for systems supporting scientific reasoning or clinical advice (Bélisle-Pipon 2024; Giorgino et al. 2023).

Safeguards such as retrieval-augmented generation (RAG) (Chen et al., 2024), expert-led evaluation, and alignment methods like reinforcement learning from human feedback (RLHF) (Christiano et al., 2023) exist, but are inconsistently applied. Many published evaluations still rely heavily on lexical overlap metrics.

While prior work notes the risks of hallucination and the limits of automatic metrics, few studies have examined these failures in domain-specific QA. Our work addresses this gap through targeted failure analyses and by proposing ethically grounded evaluation principles centered on factuality, completeness, and verifiable grounding, better reflecting the needs of high-stakes scientific tasks.

4 Methodology

To investigate the ethical limitations of automatic evaluation metrics in domain-specific question answering (QA), we analyze a French-language QA system fine-tuned on botanical texts. The system follows a two-stage architecture: the first stage generates trait-specific questions from floristic descriptions, and the second stage answers those questions based on the same context.

In our experiments, we evaluate only the answer generation stage, as factual reliability and completeness are most critical for downstream use. Question generation is included to relieve users from formulating queries and to provide standardized prompts, but its intrinsic quality was not separately assessed.

The QA system is based on LLaMA 2[2] and LLaMA 3[3] models, fine-tuned using Low-Rank Adaptation (LoRA) (Hu et al., 2021). The training dataset consists of 16,962 expert-verified question–answer pairs constructed from unstructured botanical descriptions. Each QA pair is associated with a specific botanical trait (e.g., leaf shape, flower color, inflorescence length) and was designed to reflect structured knowledge retrieval from naturalistic text.

4.1 Evaluation Dataset

For evaluation, we curated a held-out test set of 1,697 botanical contexts from a distinct source corpus not used during training. From this, a representative sample of 100 model outputs was randomly selected using a reproducible pandas-based function for expert-based review. Each sample consists of a botanical description (context), a trait-specific question, and the system-generated answer

4.2 Human Evaluation Protocol

Each of the 100 outputs was independently reviewed by a biodiversity expert. The expert rated each answer on a 1-5 Likert scale across five dimensions: Accuracy, Completeness, Relevance, Fluency, and Terminology Usage (Table 1).

[2]https://huggingface.co/meta-llama/Llama-2-7b

[3]https://huggingface.co/meta-llama/Meta-Llama-3-8B

Metric	Meaning
Accuracy	Factual correctness of the response
Completeness	Inclusion of all relevant information from the context
Relevance	Appropriateness of the answer given the question
Fluency	Grammatical and stylistic quality
Terminology Usage	Correct and domain-appropriate terminology

Table 1: Expert evaluation metrics for assessing response quality in domain-specific QA.

The expert was provided with the meaning of each evaluation expert-based metric to ensure consistent scoring in all examples.

4.3 Automatic Metrics

To assess how standard metrics reflect answer quality, we computed the following scores for the same 100 examples: BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), BERTScore (Zhang et al., 2020), and Exact Match (EM) (Rajpurkar et al., 2016a). These metrics were computed using the model outputs and their corresponding reference responses from the training set. We then compared these automatic scores to human ratings in order to analyze discrepancies and identify failure cases that raise ethical concerns.

4.4 Ethical Framing

This methodology is designed to reveal how surface-level metrics such as BLEU and ROUGE may produce inflated scores for outputs that are fluent, but factually incorrect, incomplete, or misleading. In scientific domains such as botany, such evaluation gaps pose real risks, including the propagation of inaccurate species descriptions, misclassifications of traits, and loss of trust in automated systems. By pairing automatic metrics with domain-expert assessment, our aim is to identify evaluation failures that have ethical implications for the deployment of LLMs in high-stakes QA tasks.

5 Results and Analysis

5.1 Quantitative Overview of Expert Ratings

We first report the average scores assigned by the domain expert across the five evaluation dimensions. As shown in Table 2, the model achieves high average ratings in Accuracy (4.74), Botanical Terminology Usage (4.78), and Completeness (4.53), with slightly lower but still strong scores for Relevance and Fluency (both at 4.48).

To assess whether surface-level qualities such as fluency are indicative of factual correctness, we

Metric / Expert Dimension	Mean Score
BLEU	51.48
ROUGE-L	77.13
Exact Match (EM)	0.22
BERTScore F1	0.93
Expert Accuracy	4.74
Expert Completeness	4.53
Expert Relevance	4.48
Expert Fluency	4.48
Botanical Terminology Usage	4.78

Table 2: Comparison of average automatic metric scores with expert evaluation ratings (scale: BLEU/ROUGE in %, EM in [0–1], Experts in [1–5]).

computed the Pearson correlation coefficients between the expert-rated dimensions (Table 3). Accuracy and Completeness show a moderate correlation ($r = 0.52$), while Fluency correlates only weakly with Accuracy ($r = 0.35$) and even less with Completeness ($r = 0.17$). The weakest correlation is between Relevance and Terminology ($r = 0.08$), and Fluency shows only a modest link to Botanical Terminology, yet the highest compared to other expert-based metrics ($r = 0.42$). These findings suggest that well-written outputs are not reliable indicators of factual quality.

	Acc.	Comp.	Rel.	Flu.	Term.
Accuracy	1.00	0.52	0.52	0.35	0.16
Completeness		1.00	0.30	0.17	0.12
Relevance			1.00	0.17	0.08
Fluency				1.00	0.42
Terminology					1.00

Table 3: Pearson correlation matrix between expert evaluation dimensions.

5.2 Automatic Metrics vs. Expert Ratings

To evaluate how well commonly used automatic metrics align with expert human judgment, we compared BLEU, ROUGE-L, Exact Match (EM), and BERTScore against five expert-evaluated dimensions: Accuracy, Completeness, Relevance, Fluency, and Botanical Terminology Usage. Table 2 reports the mean scores across 100 evaluated samples.

Context: Twining herbaceous plant with alternate paripinnate leaves. Rachis 7 to 9 cm long, bearing 10 to 13 pairs of rectangular leaflets 15 to 20 mm long, 6-8 mm wide, decreasing in size towards the base: apex and base almost truncated. Pairs of leaflets spaced 4-7 mm apart. Upper side of the blade glabrous, underside with fairly dense semi-erect white hairs. Petiole 2 to 15 mm long before the first pair of leaflets; rachis with fairly dense semi-erect red hairs: stems with the same pubescence.
Question: *What is the color of the hairs on the underside of the leaflets?*
Fine-tuned LLM Answer: "The underside of the leaflets is covered with semi-erect, fairly dense white hairs."
Reference Answer: The hairs are white.
Expert Evaluation: Accuracy: 5, Completeness: 5, Fluency: 5
Metrics: BLEU: 1.7, EM: 0, ROUGE-L: 22.22, BERTScore: 0.80
Comment: Correct and informative answer; metrics penalized due to elaboration.

Context: Small tree, 10 to 15 m tall, or a sarmentose-lianaceous shrub with alternate leaves. Obovate leaf blade, 8 to 18 cm long, or more, 5 to 9 cm wide at the top. Wedge-shaped base, broadly rounded top, sometimes truncated, or with a short, sharp point in the middle. 7 to 9 lateral veins and, between them, nervils spaced 3-4 mm apart, parallel to each other.
Question: *What is the base of the leaves like?*
Fine-tuned LLM Answer: "The base of the leaves is wedge-shaped."
Reference Answer: The base of the leaves is wedge-shaped, the apex is broadly rounded or sometimes truncate, or with a short abrupt point in the middle.
Expert Evaluation: Accuracy: 5, Completeness: 5, Fluency: 4
Metrics: BLEU: 8.23, EM: 0, ROUGE-L: 46.67, BERTScore: 0.84
Comment: Accurate and concise, but metrics suffer due to omission of less relevant details.

Context: Petiole 1 to 5 cm long, covered with the same whitish pubescence, as well as the stem and the 2 filiform stipules 3 to 6 mm long. Axillary flowers, isolated, or in pairs, at the top of a peduncle 1 to 2 cm long. Golden yellow corolla 20 mm wide, with 5 obtriangular petals, with a broad and emarginate apex. In the center, a short column bearing numerous stamens. Calyx with 5 fairly wide triangular teeth at the base.
Question: *Which organs are mentioned in the plant description?*
Fine-tuned LLM Answer: "The mentioned organs are the petiole, stem, stipules, flowers, corolla, petals, column, stamens, and calyx."
Reference Answer: The mentioned organs are the petiole, stem, stipules, corolla, column, stamens, calyx, and peduncle.
Expert: Accuracy: 5, Completeness: 4, Fluency: 5
Metrics: BLEU: 50.59, EM: 0, ROUGE-L: 80.0, BERTScore: 0.96
Comment: Nearly perfect, missing only the peduncle; EM fails despite strong factual match.

Table 4: Examples showing mismatch between automatic metrics and expert evaluation.

Discrepancies Between Metrics and Expert Judgment: As illustrated in Table 4, even complete and accurate factual answers can receive low automatic scores due to linguistic variation or partial overlap with reference. Despite high expert ratings for accuracy and terminology usage, Exact Match (EM) remains extremely low at 0.22. This illustrates the inadequacy of EM in settings where paraphrasing and linguistic variation are common. Similarly, BLEU (51.48) and ROUGE-L (77.13) (Table 2) reflect moderate overlap but remain insensitive to omissions or hallucinations, two critical failure modes in scientific QA.

Semantic vs. Factual Fidelity: BERTScore F1 (0.93) more closely tracks expert evaluations, suggesting better alignment with semantic content. However, BERTScore cannot distinguish between correct information and plausible-sounding hallucinations, nor does it penalize factual incompleteness. These results reinforce the notion that semantic similarity does not imply factual fidelity.

Ethical Implications: These discrepancies raise serious ethical concerns. In high-stakes domains like biodiversity, law, and medicine, models can receive strong automatic scores while omitting crucial details or introducing unverifiable content. Therefore, over-reliance on surface-level metrics can mislead downstream users, researchers, or policymakers into trusting outputs that lack scientific rigor.

We provide empirical evidence for the core claim of this paper: that standard metrics such as BLEU, ROUGE, EM, and BERTScore fail to capture the factual quality of LLM-generated answers in domain-specific settings. We argue for incorporating expert validation and task-specific evaluation frameworks as ethical imperatives in future work on domain-adapted QA systems.

5.3 Alignment of Automatic Metrics with Expert Ratings

To further quantify how well automatic metrics track expert judgment, we computed Pearson correlations between BLEU, ROUGE-L, EM, and

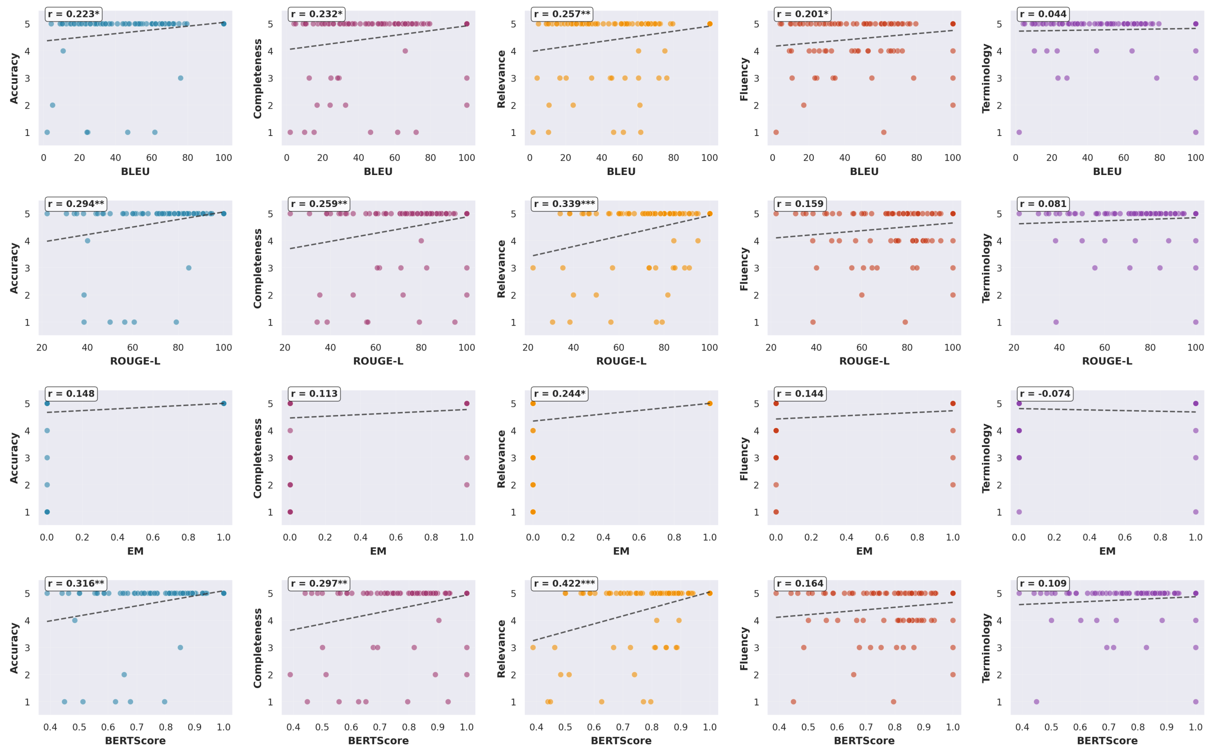


Figure 1: Scatter plot matrix showing automatic metrics vs. expert evaluation dimensions (1–5 scale). Pearson correlations shown with significance levels (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$). n=100.

BERTScore and the five expert-rated dimensions (Accuracy, Completeness, Relevance, Fluency, Terminology). Figure 2 visualizes the results.

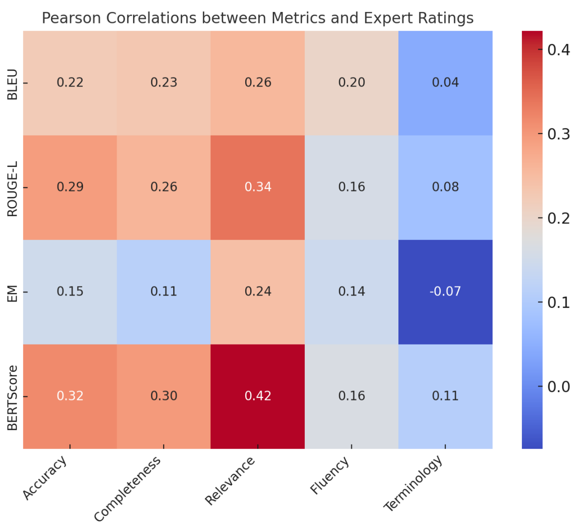


Figure 2: Pearson correlations between automatic metrics (rows) and expert-rated dimensions (columns). Values indicate weak-to-moderate alignment at best; EM is largely uninformative, and BERTScore correlates most with Relevance rather than factual Accuracy.

Overall, alignments are weak to moderate. The strongest association is BERTScore with Relevance ($r \approx 0.42$), followed by ROUGE-L with Relevance ($r \approx 0.34$). Correlations with Accuracy are only modest across metrics (BLEU ≈ 0.22, ROUGE-L ≈ 0.29, BERTScore ≈ 0.32), and associations with Fluency are uniformly low ($r \leq 0.20$). Terminology exhibits the weakest alignment overall (e.g., EM ≈ -0.07). Notably, Exact Match is effectively uncorrelated with all expert dimensions, showing its weakness when faced with paraphrased or partially correct answers. These findings reinforce that surface-similarity metrics (BLEU, ROUGE. EM) and even semantic similarity (BERTScore) do not reliably capture factual correctness, completeness, or terminological precision in domain-specific QA.

Furthermore, the correlation analysis (Figure 1) reveals significant limitations in current automatic evaluation metrics for botanical QA assessment. BERTScore demonstrates the strongest alignment with expert judgments, showing moderate correlations with semantic dimensions: Relevance ($r = 0.422^{***}$), Accuracy ($r = 0.316^{**}$), and Completeness ($r = 0.297^{**}$). ROUGE-L exhibits weaker but statistically significant correlations, particularly with Relevance ($r = 0.339^{***}$) and Completeness ($r = 0.259^{**}$). BLEU shows minimal correlations across all dimensions ($r \leq 0.257$), with only Relevance reaching significance ($r = 0.257^{**}$).

Exact Match proves largely uninformative with weak correlations ($r \leq 0.244$) and limited significance. Critically, all automatic metrics show negligible correlations with Terminology assessment ($r \leq 0.109$, mostly non-significant), which highlights their inability to capture domain-specific linguistic accuracy crucial for specialized QA systems. The moderate correlations overall (highest $r = 0.422$) indicate that automatic metrics capture only partial aspects of expert-valued quality, with BERTScore being the most reliable predictor, while human evaluation remains essential for comprehensive assessment in specialized domains.

6 Conclusion

We highlight the limitations of widely used automatic evaluation metrics: BLEU, ROUGE, Exact Match, and BERTScore in capturing the factual accuracy, completeness, and domain-specific fidelity of LLM-generated answers in scientific question answering. Our comparative analysis against expert ratings reveals that these metrics often reward superficial overlap while failing to penalize critical omissions, hallucinations, or terminological imprecision.

We argue that relying solely on these metrics can lead to misleading conclusions about model performance, particularly in high-stakes fields such as biodiversity. As illustrated through both aggregate scores and specific examples, expert-based evaluation provides a more reliable lens for assessing output quality in domain-adapted QA systems.

Future work should prioritize the development of evaluation frameworks that integrate domain expertise, task-specific criteria, and human-in-the-loop feedback. Doing so is not only methodologically sound but ethically necessary to ensure the safe deployment of LLMs in scientific and ecological applications.

7 Limitations

While our analysis highlights important shortcomings of automatic evaluation metrics in domain-specific QA, several limitations remain.

First, our study focuses on a single domain, botanical and ecological question answering using a dataset of 100 expert-rated examples. Although the findings are indicative, they may not fully generalize, to the same degree, to all other scientific or technical fields with different terminological structures or reasoning demands.

Second, expert evaluation, while more reliable than surface-level metrics, introduces its own subjectivity. Although we employed a biodiversity expert with domain knowledge, future work should include multiple annotators to assess inter-annotator agreement.

Third, our evaluation primarily addresses short-form, extractive QA responses. Longer, multi-step, or generative answers may pose different challenges, particularly around discourse coherence, reasoning chains, and multi-document grounding areas not fully captured in our current setup.

Finally, we did not explore recent or emerging evaluation methods such as LLM-as-a-judge or retrieval-augmented verification, which may complement expert-based evaluation or improve factuality assessment in future iterations.

Addressing these limitations in future work will be critical to building more robust and trustworthy evaluation pipelines for domain-adapted QA systems.

Acknowledgments

This work is co-funded by the European Union's Horizon Europe research and innovation program Cofund SOUND.AI under the Marie Sklodowska-Curie Grant Agreement No 101081674. This research is also part of the e-COL+ project (ANR-21-ESRE-0053).

References

Youssef Al Mouatamid, Jihad Zahir, Marie Bonnin, and Hajar Mousannif. 2023. *Assessing Ocean's Legal Protection Using AI: A New Dataset and a BERT-Based Classifier*.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024a. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024b. Learning from mistakes makes llm better reasoner.

Jean-Christophe Bélisle-Pipon. 2024. Why we need to be careful with llms in medicine. *Frontiers in Medicine*, Volume 11 - 2024.

Andry Castro, João Pinto, Luís Reino, Pavel Pipek, and César Capinha. 2024. Large language models overcome the challenges of unstructured text data in ecology. *Ecological Informatics*, 82:102742.

X. Chen, L. Wang, M. You, W. Liu, Y. Fu, J. Xu, S. Zhang, G. Chen, K. Li, and J. Li. 2024. Evaluating and enhancing large language models' performance in domain-specific medicine: Development and usability study with docoa. *Journal of Medical Internet Research*, 26:e58158.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences.

Hao Cui, Zahra Shamsi, Gowoon Cheon, Xuejian Ma, Shutong Li, Maria Tikhanovskaya, Peter Norgaard, Nayantara Mudur, Martyna Plomecka, Paul Raccuglia, Yasaman Bahri, Victor V. Albert, Pranesh Srinivasan, Haining Pan, Philippe Faist, Brian Rohr, Ekin Dogus Cubuk, Muratahan Aykol, Amil Merchant, Michael J. Statt, Dan Morris, Drew Purves, Elise Kleeman, Ruth Alcantara, Matthew Abraham, Muqthar Mohammad, Ean Phing VanLee, Chenfei Jiang, Elizabeth Dorfman, Eun-Ah Kim, Michael P Brenner, Viren Jain, Sameera Ponda, and Subhashini Venugopalan. 2025. Curie: Evaluating llms on multitask scientific long context understanding and reasoning.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Filip Dorm, Joseph Millard, Drew Purves, Michael Harfoot, and Oisin Mac Aodha. 2025. Large language models possess some ecological knowledge, but how much? *bioRxiv*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-controlled alpacaeval: A simple way to debias automatic evaluators.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization.

Md Meftahul Ferdaus, Mahdi Abdelguerfi, Elias Ioup, Kendall N. Niles, Ken Pathak, and Steven Sloan. 2024. Towards trustworthy ai: A review of ethical and robust large language models.

Riccardo Giorgino, Mario Alessandri-Bonetti, Andrea Luca, Filippo Migliorini, Nicolò Rossi, Giuseppe M. Peretti, and Laura Mangiavini. 2023. Chatgpt in orthopedics: a narrative review exploring the potential of artificial intelligence in orthopedic practice. *Frontiers in Surgery*, Volume 10 - 2023.

Andrew V. Gougherty and Hannah L. Clipp. 2024. Testing the reliability of an ai-based large language model to extract ecological information from the scientific literature. *npj Biodiversity*, 3(1):13.

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.

Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975, Miami, Florida, USA. Association for Computational Linguistics.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M. Krumholz, Jure Leskovec, Eric J. Topol, and Pranav Rajpurkar. 2023. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265.

Ayoub Nainia, Régine Vignes-Lebbe, Eric Chenin, Maya Sahraoui, Hajar Mousannif, and Jihad Zahir. 2024. Floraner: A new dataset for species and morphological terms named entity recognition in french botanical text. *Data in Brief*, 56:110824.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. 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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. 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.

Ravi Shanker Raju, Swayambhoo Jain, Bo Li, Jonathan Lingjie Li, and Urmish Thakker. 2024. Constructing domain-specific evaluation sets for LLM-as-a-judge. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 167–181, Miami, Florida, USA. Association for Computational Linguistics.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022a. Galactica: A large language model for science.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022b. Galactica: A large language model for science.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

From Zero to Hero: Building Serbian NER from Rules to LLMs

Milica Ikonić Nešić
University of Belgrade, Faculty of Philology, Serbia
milica.ikonic.nesic@fil.bg.ac.rs

Saša Petalinkar
University of Belgrade, Serbia
sasa5linkar@gmail.com

Ranka Stanković
University of Belgrade, Faculty of Mining and Geology, Serbia
ranka.stankovic@rgf.bg.ac.rs

Ruslan Mitkov
University of Alicante, Spain
ruslan.mitkov@ua.es

Abstract

Named Entity Recognition (NER) presents specific challenges in Serbian, a morphologically rich language. To address these challenges, a comparative evaluation of distinct model paradigms across diverse text genres was conducted. A rule-based system (SrpNER), a traditional deep learning model (Convolutional Neural Network – CNN), fine-tuned transformer architectures (Jerteh and Tesla), and Large Language Models (LLMs), specifically ChatGPT 4.0 Nano and 4.1 Mini, were evaluated and compared. For the LLMs, a one-shot prompt engineering approach was employed, using prompt instructions aligned with the entity type definitions used in the manual annotation guidelines. Evaluation was performed on three Serbian datasets representing varied domains: newspaper articles, history textbook excerpts, and a sample of literary texts from the srpELTeC collection. The highest performance was consistently achieved by the fine-tuned transformer models, with F1 scores ranging from 0.78 on newspaper articles to 0.96 on primary school history textbook sample.

1 Introduction

The task of Named Entity Recognition (NER) involves identifying and classifying key information, such as persons, locations, organisations, dates, and other specific entities, within unstructured text (Krstev et al., 2014; Frontini et al., 2020). Accurate NER is crucial for numerous downstream NLP applications, including information extraction (Feng et al., 2022), question answering (Mollá et al., 2006; Verma et al., 2023), and machine translation (Sulistyo et al., 2025). Historically, NER systems have evolved through various approaches, ranging from rule-based methods to those leveraging machine learning and deep learning. Rule-based systems, exemplified by SrpNER for the Serbian language (Krstev et al., 2014), which utilised extensive lexical resources (Krstev, 2008; Vitas and Krstev, 2012) and local grammars, demonstrated high efficiency, particularly on specific text types like news articles (achieving an F1 score of approximately 96% on newspaper texts). However, their development necessitates significant linguistic expertise, and adapting them to new classes, domains or languages can be resource-intensive.

Moving beyond rule-based systems, machine learning and deep learning approaches, including models like Conditional Random Fields, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs), became prevalent. While demonstrating impressive results for high-resource languages with extensive datasets, these models also showed potential for more specific or challenging contexts. For example, CNN architectures have been successfully employed for NER tasks in specific low-resource domains, such as legal text in Turkish (Çetindağ et al., 2023) or historical literary text in Serbian (Šandrih Todorović et al., 2021), achieving competitive performance (F1 scores aprox 91%) on such datasets.

The advent of transformer-based models marked a significant paradigm shift in NLP. BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) emerged as a cornerstone model, setting new benchmarks across a wide array of language understanding tasks. Its architecture enables the learning of deep contextualised representations, leading to superior performance in tasks like NER (Zhang and Zhang, 2023). BERT's capability for transfer learning has proven particularly beneficial for low-shot classification tasks (Garrido-Merchan et al., 2023). The success of BERT spurred the development of multilingual models (Wang et al., 2020) and specialised models for various non-English languages. For South Slavonic languages, dedicated models like Bertić (Lju-

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 87–96, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-010

bešić and Lauc, 2021), xlm-r-bertić [1] developed by the CLARIN Knowledge Centre for South Slavonic languages (CLASSLA), SRoBERTa (Cvejić, 2022), and XLM-R based models jerteh-355-tesla (Ikonić Nešić et al., 2024) and TESLA-mini (Škorić, 2024) have been developed, demonstrating the effectiveness of the transformer architecture in this linguistic context.

More recently, Large Language Models (LLMs) (Brown et al., 2020) have demonstrated remarkable zero-shot and few-shot abilities across numerous NLP tasks via prompt engineering (Li and Liang, 2021). This approach allows leveraging the vast knowledge within frozen pre-trained LLMs by crafting specific input prompts, appealing greatly to low-resource scenarios as it avoids resource-intensive training. However, applying prompt learning directly to tasks like NER presents unique challenges (Shen et al., 2023). While LLMs excel at tasks aligned with their pre-training objectives (e.g., text generation or "fill-in-the-blank"), NER is fundamentally a sequence labelling task requiring precise identification of entity spans and types (Ma and Hovy, 2016). Early prompt-based NER approaches, such as span-orientated methods that enumerate all potential spans (Cui et al., 2021) or type-oriented methods that query for specific entity types (Liu et al., 2022), often required multiple inference rounds or relied on complex, hand-crafted prompt templates, limiting their efficiency and practical applicability (Shen et al., 2023). This inherent mismatch means that despite the general capabilities of LLMs, achieving robust and accurate NER performance via prompt engineering is still an active area of research and often requires careful prompt design or specialised techniques.

This paper presents a comparative analysis of the performance of different generations of models applied to NER in Serbian. The comparison of rule based, traditional deep learning approaches, represented by a trained CNN model, with two fine-tuned BERT models is presented. Furthermore, the potential of leveraging contemporary LLMs for Serbian NER through prompt engineering, utilising the capabilities of the ChatGPT 4.0 mini and ChatGPT 4.0 nano models was investigated. By evaluating and comparing these diverse modelling paradigms which is a major contribution of the this study, we aim to provide insights into their relative strengths, weaknesses, and applicability for Serbian NER, contributing to the understanding of model evolution and resource efficiency in this field.

2 Related Work

While NER has a long research history, its comparative evaluation across model generations remains unevenly distributed across languages. In particular, Serbian has seen a few dedicated surveys or systematic comparisons only. The earliest and most relevant work is by Vitas and Pavlović-Lažetić (2008), who provided a general overview of NER methods and linguistic resources for Serbian, including rule-based systems. Since then, most research has focused on domain-specific applications or evaluated individual systems. For instance, Sandrih et al. (2019) examined NER systems for Serbian personal names, while Todorović et al. (2021) developed models for recognising entities in 19th-century Serbian literature. More recently, Živković et al. (2022) assessed transformer-based models in the clinical domain. However, none of these works offer a broad, comparative evaluation across diverse model paradigms, nor do they address resource-efficiency concerns across domains.

In contrast, surveys on NER for English are abundant and continuously updated. Well-cited foundational works such as Nadeau and Sekine (2007), Marrero et al. (2013), and Shaalan (2014) laid the groundwork. More recently, deep learning–focused surveys like Li et al. (2020), Keraghel et al. (2024), and Warto et al. (2024) have provided extensive reviews of neural and transformer-based NER systems. Domain-specific surveys also exist, such as Ehrmann et al. (2023) for historical texts and Jehangir et al. (2023) for biomedical and multilingual NER. This disparity further motivates our study, which addresses a clear gap in the literature for Serbian and offers a multi-paradigm evaluation from rule-based through deep learning to LLM-based NER systems.

The study by (Affi and Latiri, 2022) addresses NER for the Arabic language, highlighting the challenges posed by its complex morphology which often necessitates extensive handcrafted feature engineering. To overcome this limitation, the authors proposed a novel deep neural network architecture combining CNN, LSTM, and BERT embeddings to generate rich word representations without re-

[1]Classla/xlm-r-bertic · Hugging Face. (2023, December 18). https://huggingface.co/classla/xlm-r-bertic

lying on external knowledge or handcrafted features. Their approach achieved state-of-the-art results on the ANERCorp dataset, with F1-scores of 93.34% and 93.68% using bidirectional LSTM-CRF (BLC) and bidirectional GRU-CRF (BGC) architectures, respectively. This work is relevant to our study as it demonstrates the effectiveness of advanced deep learning architectures (integrating embeddings from models like BERT with sequence models like LSTM/GRU and CRF) for NER, even for morphologically rich languages. Furthermore, it implicitly includes a comparison of the performance between related architectures (LSTM vs. GRU) within their proposed framework, which aligns with our goal of comparing different model types.

Shelar et al. (2020) conduct a comparative analysis of different existing libraries and tools for NER, including Python's spaCy, Apache OpenNLP, and TensorFlow. The comparison was based on key performance metrics such as training accuracy, F-score, prediction time, model size, and ease of training, using the same dataset across all evaluated tools. A key finding was that Python's spaCy generally achieved higher accuracy and better overall results compared to the other tools tested. This paper is highly relevant to our research as it serves as a direct example of a comparative study of different NER systems or implementations. Its methodology of using standard performance metrics to evaluate distinct tools provides a valuable template and context for our own comparison of various NER models, even if our focus might be more on the underlying model architectures rather than solely the libraries used.

The research presented in (Ikonić Nešić et al., 2024) investigates NER for the Serbian, focusing on the integration of BERT models with the spaCy library. The paper presents a comparison of different architectures and techniques for preparing NER models, trained to recognise seven entity types on a diverse Serbian dataset. Specifically, the authors explored various configurations and training pipelines within the spaCy framework, as well as the impact of different BERT versions (varying architectures, sizes, and pre-training corpora containing Serbian). The goal was to evaluate the trade-offs between model complexity and performance. This research is relevant as it addresses NER for a specific language (Serbian), which is the focus of our study. Most importantly, the paper explicitly conducts a comparison of different configurations and variations of a powerful NER approach (BERT+spaCy), analysing their impact on performance, which relates to our objective of comparing different NER models or different configurations/implementations.

The current study provides a comprehensive comparative analysis of NER models for the Serbian, spanning rule-based systems, traditional machine learning approaches, modern deep learning architectures, and LLMs. In contrast to prior work, which has typically focused on specific domains or isolated model types, our evaluation is conducted across multiple real-world text genres including historical textbooks, news articles, and literary prose, allowing for a robust assessment of model generalisation and domain adaptability. By systematically benchmarking diverse NER paradigms on both seen and unseen data, we offer novel insights into the strengths and limitations of each approach, thereby contributing to the advancement of NER in low-resource and morphologically rich languages.

3 Methodology

This section outlines the dataset preparation process for model training, the training of CNN, two BERT-based models, as well as the one-shot prompting approach applied to LLMs.

3.1 Data Preparation

The preparation of the training dataset has been ongoing for an extended period and constitutes a part of the TESLA-NER-NEL corpus (Ikonić Nešić and Utvić, 2024), which, in its final version, will contain 150,000 sentences annotated with named entities linked to Wikidata entries, as well as part-of-speech (POS) tags and lemmatisation.

The training dataset was compiled through two distinct annotation strategies: a semi-automated procedure (*srTESLA-SA*) and a fully automated one (*srTESLA-FA*). Within the semi-automated workflow, a total of 53,417 sentences were initially labelled automatically using SrpNER (Krstev et al., 2014) and jerteh-355-tesla (Ikonić Nešić et al., 2024) model. For manual correction of pre-annotated dataset, INCEpTION tool (Figure 1) was used. The sentences were post-annotated by multiple trained annotators, and all annotations were cross-checked by an expert. These sentences were selected from (1) novels from SrpELTeC (Stanković et al., 2024) and SrpKor (Vitas et al., 2024)

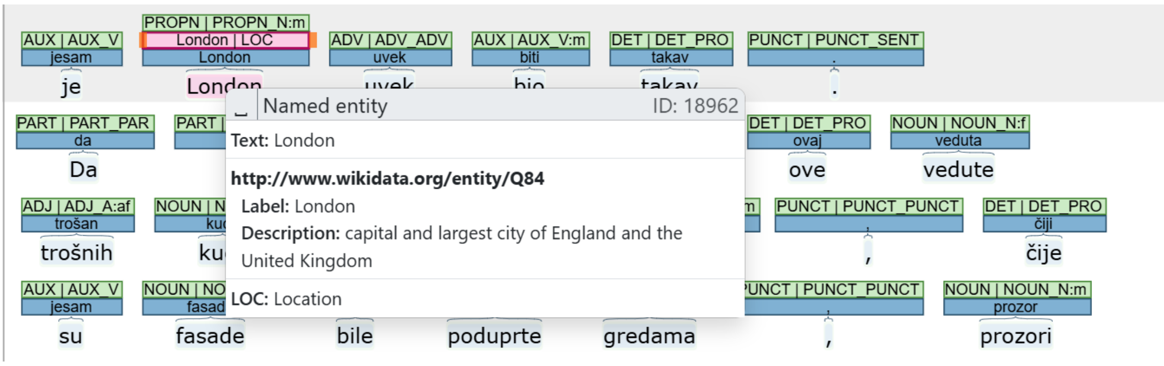


Figure 1: An example of annotation in INCEpTION

(23,273 sentences), (2) newspapers from SrpKor (8,737 sentences), (3) legal documents from Intera (Stanković et al., 2017) (19,383 sentences) and (4) wikipedia from srELEXIS (Krstev et al., 2024) (2024 sentences).

The fully automated approach relied on two techniques: the first utilised sentence templates and structured lexical resources, including gazetteers such as Leximirka (Stanković et al., 2018; Lazic and Škoric, 2019), to generate annotated examples (*srTESLA-lex*); the second employed ChatGPT 4.0 for automatic annotation generation (*srTESLA-chat*). This approach provided context-rich sentences, facilitating disambiguation for NEL task, with 20,076 sentences in total.

The named entity tagset used in this study is aligned with categories commonly applied in the annotation of literary and historical texts, such as those developed within the European Literary Text Collection (ELTeC) (Stanković et al., 2024; Frontini et al., 2020). It includes the following entity types: personal names (PERS), geographical locations (LOC), organizations (ORG), professional roles and titles (ROLE), demonyms (DEMO), cultural and artistic works (WORK), and events (EVENT). Among these, locations (LOC) are the most frequent, with approximately 36,654 instances, followed by personal names (PERS) with 13,636, and organization names (ORG) with around 11,060 occurrences (Table 1).

3.2 NER Modelling Approaches and Configuration

The configuration and implementation of the NER task across different modelling approaches is presented in this subsection.

One-Shot LLM-based NER with `spacy-llm`

To explore the capabilities of modern LLMs for Serbian NER, we employed a one-shot learning strategy facilitated by the `spacy-llm` [2] library. This approach avoids resource-intensive fine-tuning by leveraging the model's existing knowledge through carefully crafted prompts.

Recognising that prompt performance is often enhanced in multilingual contexts and by providing concrete examples, we designed a custom prompt template using the `Jinja` [3] templating language. This allowed for a dynamic and structured input for the LLMs. The core of our prompt is designed to instruct the model to act as an expert in NER and to identify entities within a given text according to a specified set of categories.

The `Jinja` template is structured as follows:

```
Vi ste stručnjak za prepoznavanje
    imenovanih entiteta (NER).
Vaš zadatak je da primite tekst i iz
    njega izdvojite imenovane entitete.
Svaki entitet mora pripadati jednoj od
    sledećih kategorija:
    {{ ', '.join(labels) }}.
Ako neki deo teksta nije entitet, označ
    ite ga kao: `==NONE==`.

{%- if label_definitions %}
Ispod su definicije svake kategorije
    koje će vam pomoći da tačno
prepoznate vrste imenovanih entiteta.
{%- endif %}
...
Pasus: {{ text }}
Odgovor:
```

The NER task was configured within the `spacy-llm` framework by defining the set of entity labels and providing their detailed description specified below.

[2] https://spacy.io/usage/large-language-models
[3] https://jinja.palletsprojects.com/en/stable

Class	Description for annotators	Count
LOC	Names of continents, countries, populated places, geographic features, celestial bodies, etc.	36,654
ROLE	Professional titles, functions, or social roles, such as doctor, director, king, or teacher.	15,170
PERS	Personal names of individuals, including given names, surnames, and aliases of real or fictional figures.	13,636
ORG	Names of institutions, companies, political bodies, schools, hospitals, and other formal organizations.	11,060
DEMO	Demonyms indicating origin, nationality, or ethnic background, including adjectival forms derived from locations.	7,559
WORK	Titles of creative works such as books, poems, artworks, theatrical plays, and periodicals.	3,319
EVENT	Specific historical or recurring events such as wars, revolutions, natural disasters, or commemorations.	464

Table 1: Entity types with descriptions and frequency in the dataset.

```
[components.llm.task.label_definitions]
PERS = "Vlastita imena stvarnih ili izmi
    šljenih pojedinaca   li  čna imena,
    prezimena, nadimci, bogovi, sveci i
    imenovane životinje."
ROLE = "Zanimanja, činovi, titule i
    funkcije koje ljudi obavljaju, sa
    ili bez ličnog imena; uključuje viš
    erečna zvanja."
DEMO = "Nazivi naroda, etničkih grupa i
    stanovnika mesta, kao i pridevi
    izvedeni iz geografskih imena."
ORG = "Imena organizacija, institucija i
     udruženja: kompanije, partije, š
    kole, muzeji, kafane, crkve,
    sportski k l u b o v i "
LOC = "Vlastita imena geografskih
    lokacija: kontinenti, države,
    regioni, gradovi, sela, planine,
    reke, jezera, ulice, trgovi."
WORK = "Naslovi umetničkih i kulturnih
    dela: knjige, pesme, filmovi, slike,
     skulpture, spomenici, novine, video
    -igre."
EVENT = "Nazivi događaja: praznici,
    revolucije, ratovi, bitke,
    demonstracije, festivali, sportski
    događaji, prirodne katastrofe."
```

For these experiments, two specific models were employed: `gpt-4.1-mini` and `gpt-4.1-nano`. These models were tasked with performing NER on our evaluation datasets using the described one-shot prompting configuration.

CNN and Fine-Tuned Transformer Models

In addition to the LLM-based prompting method, we trained a CNN and two transformer-based models to serve as comparative baselines. These experiments were conducted within the `spaCy` framework, making use of the core library for the CNN and the `spacy-transformers` extension for the transformer models.

The **CNN model** was configured using `spaCy`'s standard multi-layer `tok2vec` architecture. This component generates context-sensitive token vectors which are then passed to the Named Entity Recognition (`ner`) layer for classification.

The two **transformer models** leverage the `spacy-transformers` library to integrate pretrained language models into the `spaCy` pipeline. The transformer's contextual word embeddings are fed into the `ner` component. The specific pretrained models used as a base for fine-tuning were:

- `te-sla/TeslaXLM`[4] model is derived from the large multilingual architecture, FacebookAI/xlm-roberta-large, having been further fine-tuned for the nuances of Serbian and Serbo-Croatian. Comprising 561 million parameters, its adaptation involved training on a substantial 20-billion-token corpus encompassing both Latin and Cyrillic scripts commonly used in Serbian. This comprehensive fine-tuning process results in a model that demonstrates high proficiency and robustness across varying scripts and dialectal forms (Škorić and Petalinkar, 2024).

- `jerteh/Jerteh-355`[5] is based on the RoBERTa-large architecture but was trained from scratch exclusively on a monolingual Serbian corpus of 4 billion tokens. With 355 million parameters, this model is specifically tailored to generate high-quality, context-aware embeddings optimized for the Serbian language environment (Škorić, 2024).

To ensure a fair comparison, all three models were trained on the same dataset for a total of 10 epochs for transformers and 5 epochs for CNN.

[4] https://huggingface.co/te-sla/TeslaXLM
[5] https://huggingface.co/jerteh/Jerteh-355

4 Evaluation Results and Discussion

The trained models were evaluated for their performance on the test set, as shown in Table 2. To assess performance across different text types, the models were evaluated on data from three distinct sources: **newspaper articles** from *Politika* newspapers (841 sentences), a sample from corpus of history textbook for elementary school named **srHistory** (331 sentences), and a sample of three novels from the srpELTeC collection (SRP19070[6], SRP19121[7], SRP19180[8]) (Table 3) named **srpELTeC sample** (544 sentences), were the literary texts used for evaluation were explicitly excluded from the training set. The distribution of named entities across these datasets is shown in Figure 2.

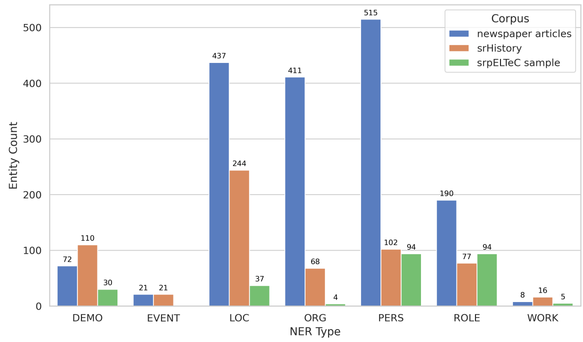


Figure 2: Distribution of NE types across corpora

The models included in the comparison represent distinct paradigms: the rule-based `SrpNER`, a traditional `CNN`, fine-tuned BERT models (`Jerteh` and `Tesla`), and prompt-based LLMs (`ChatGPT 4.0 Nano` and `ChatGPT 4.1 Mini`). As anticipated, the fine-tuned transformer models `Jerteh` and `Tesla` generally achieved the highest overall F1 scores across the evaluation sets (Table 3). Their strong performance stems from their ability to learn complex contextual patterns from the large and diverse training dataset as TESLA-NER-NEL and generalise these patterns. The rule-based SrpNER system demonstrated robustness, performing strongly on domains it was specifically designed for, such as newspaper articles, and maintaining solid performance on other domains due to its reliance on linguistic rules and lexicons, although it exhibits less flexibility than machine learning models when encountering entirely new patterns and types (e.g. EVENT). The `CNN`, a traditional deep learning approach, proved more susceptible to domain and style shifts; while capable of learning effective local features, it is less adept at capturing long-range context compared to transformers, which can hinder performance on complex sentences or subtle entity mentions. The prompt-based `LLMs` approach show a notable decrease in performance on the more challenging or domain-shifted datasets (Table 3). A key aspect of LLM evaluation approach involved providing the `ChatGPT` models with detailed instructions and definitions for each entity type directly in the prompt. These instructions were identical to those provided to human annotators who created the gold standard dataset used for training and evaluation. This consistency ensures that both the LLMs and the supervised models are attempting to solve the exact same NER task definition. By leveraging the LLMs' strong instruction-following capabilities with the annotation guidelines, we aimed to facilitate a direct and fair comparison between the performance achieved via prompt engineering and that of models explicitly trained on data annotated according to those same guidelines. Despite this, the inherent nature of prompt-based generation, as opposed to fine-tuned sequence labelling, appears less optimal for achieving high precision and recall on this specific task without further adaptation, as detailed by their class-level results (Table 4).

Analysing performance by dataset and entity type reveals the impact of textual characteristics (Table 4). The *srHistory* dataset has very strong results from BERT models (`Tesla` F1 0.958, `Jerteh` F1 0.884 overall), and exceptional per-class performance for `Tesla`, achieving F1 scores of 0.94 or higher for most entity types, including PERS, LOC, ORG, DEMO, and EVENT (Table 4b). This high performance, combined with Tesla's low FP count on this dataset (Table 3), suggests the well known named entities and clear style of a textbook, with is well suited to its capabilities. `SrpNER` and `CNN` also showed solid per-class results on this dataset, outperforming the `LLMs` on many entity types. On the *newspapers* dataset, `Tesla`, `Jerteh` and `SrpNER` models performed well on common entity types like PERS, LOC, and ORG (Table 4a), likely because this domain aligns closely with the newspaper portion of the training data. `SrpNER` showed particularly high precision for these classes, consistent with its rule-based na-

[6] Jelena Dimitrijević, Fati-Sultan (ELTeC edition)
[7] Veljko M. Milićević, *Bespuće* (Wasteland ELTeC edition)
[8] Milica Jankovic, *Pre sreće* (Before happiness ELTeC edition)

Table 2: Precision, recall, and F1 for each entity type and model on a test dataset, including entity counts.

Class	Count	Tesla			Jerteh			CNN		
		P	R	F1	P	R	F1	P	R	F1
PERS	7,471	0.972	0.972	**0.972**	0.955	0.976	0.965	0.918	0.890	0.904
LOC	2,910	0.966	0.975	**0.971**	0.966	0.973	0.970	0.940	0.944	0.942
ROLE	2,719	0.844	0.820	**0.832**	0.825	0.837	0.831	0.795	0.756	0.775
DEMO	2,053	0.934	0.966	**0.950**	0.931	0.959	0.945	0.903	0.902	0.902
ORG	1,458	0.817	0.817	**0.817**	0.807	0.802	0.804	0.708	0.742	0.725
WORK	699	0.698	0.645	0.671	0.659	0.718	**0.687**	0.582	0.476	0.524
EVENT	100	0.736	0.670	**0.702**	0.685	0.630	0.656	0.786	0.330	0.465

Table 3: Summary of NER model performance across three distinct datasets

Dataset	Total Entities	Model	TP	FP	FN	P	R	F1
Newspaper articles	1,654	SrpNER	965	140	689	0.873	0.583	0.699
		CNN	834	613	820	0.576	0.504	0.538
		Jerteh	1,339	563	315	0.704	0.810	0.753
		Tesla	1,375	476	279	0.743	0.831	**0.785**
		Chat 4.0 Nano	1,027	735	627	0.583	0.621	0.601
		Chat 4.1 Mini	1295	944	359	0.578	0.783	0.665
srHistory	638	SrpNER	472	117	166	0.801	0.740	0.769
		CNN	487	98	151	0.833	0.763	0.796
		Jerteh	559	68	79	0.892	0.876	0.884
		Tesla	601	15	37	0.976	0.942	**0.958**
		Chat 4.0 Nano	329	251	309	0.567	0.516	0.540
		Chat 4.1 Mini	540	253	98	0.681	0.846	0.755
sprELTeC Sample	264	SrpNER	209	54	55	0.795	0.799	0.793
		CNN	98	56	166	0.636	0.371	0.469
		Jerteh	163	58	101	0.738	0.617	0.672
		Tesla	200	29	64	0.873	0.758	**0.811**
		Chat 4.0 Nano	135	338	129	0.285	0.511	0.366
		Chat 4.1 Mini	221	179	43	0.553	0.837	0.666

ture, though with lower recall on some rarer types (e.g., WORK, EVENT).

In contrast, the *srpELTeC sample* dataset presented the greatest challenge for most models, leading to a significant overall performance drop, particularly for the `CNN` (F1 0.469) and `ChatGPT 4.0 Nano` (F1 0.366). This is primarily attributable to a substantial domain and style shift (19th-century literary language vs. modern training data). A specific instance of the style shift impacting LLMs was observed with the term *Arnautin*. This archaic and historical term, originating from Turkish, was not recognised by the ChatGPT 4.1 Nano model, while other models successfully identified it, illustrating the potential sensitivity of LLMs to lexical and historical variations not prominent in their pre-training. The class-level results for ELTeC (Table 4c) clearly show this difficulty across multiple entity types for these models. For example, `ChatGPT 4.0 Nano` exhibited very low precision across several common classes (PERS, LOC, DEMO). The `CNN` also showed low F1 scores on most classes. Interestingly, the rule-based `SrpNER` demonstrated relatively more stable per-class performance on some types (e.g., ROLE F1 0.80) compared to some data-driven models (`Jerteh` ROLE F1 0.54, `ChatGPT 4.1 Mini` ROLE F1 0.60), suggesting its rules were less affected by stylistic nuances than statistical patterns learned by `CNN` or `LLMs`. Transformer models (`Tesla` F1 0.811, `Jerteh` F1 0.672 overall), while still the best performers on this challenging set, showed reduced per-class scores compared to other datasets for types like ORG, ROLE, and WORK. The `ChatGPT 4.1 Mini` model on *srpELTeC sample* showed a pattern of higher recall but lower precision compared to Tesla for some classes (e.g., PERS, LOC, DEMO), indicating it retrieved more potential entities but with more false positives.

5 Conclusion/Future Work

In this study, a comparative evaluation of diverse NER model paradigms for Serbian was conducted across distinct text genres: newspa-

Table 4: Evaluation Results by Entity Type (Precision / Recall / F1)

(a) newspaper articles

Class	SrpNER			CNN			Jerteh			Tesla			Nano			Mini		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PERS	0.92	0.67	0.77	0.76	0.55	0.64	0.89	0.92	0.90	0.91	0.91	0.91	0.65	0.78	0.71	0.89	0.87	0.88
LOC	0.85	0.86	0.86	0.67	0.67	0.67	0.83	0.87	0.85	0.86	0.90	**0.88**	0.65	0.78	0.71	0.84	0.78	0.81
ORG	0.89	0.36	0.51	0.46	0.28	0.35	0.76	0.70	0.73	0.75	0.75	**0.75**	0.60	0.46	0.52	0.58	0.73	0.65
ROLE	0.78	0.47	**0.59**	0.35	0.51	0.42	0.31	0.62	0.41	0.38	0.63	0.48	0.45	0.36	0.40	0.34	0.63	0.46
WORK	1.00	0.13	0.22	0.31	0.50	**0.38**	0.23	0.38	0.29	0.13	0.13	0.13	0.02	0.13	0.03	0.61	0.38	0.11
DEMO	1.00	0.07	0.13	0.46	0.56	0.50	0.58	0.93	0.71	0.58	0.94	**0.72**	0.29	0.31	0.30	0.20	0.96	0.33
EVENT	0.00	0.00	0.00	0.14	0.05	0.07	0.37	0.48	0.42	0.52	0.67	**0.58**	0.27	0.14	0.19	0.18	0.62	0.28

(b) srHistory

Class	SrpNER			CNN			Jerteh			Tesla			Nano			Mini		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PERS	0.77	0.83	0.80	0.85	0.89	0.87	0.99	0.98	0.99	1.00	1.00	**1.00**	0.53	0.62	0.57	0.98	0.98	0.98
LOC	0.85	0.90	0.87	0.87	0.89	0.88	0.93	0.89	0.91	0.97	0.97	**0.97**	0.71	0.76	0.73	0.89	0.90	0.89
ORG	0.61	0.34	0.43	0.61	0.37	0.46	0.80	0.77	0.78	0.94	0.94	**0.94**	0.54	0.38	0.45	0.70	0.88	0.78
ROLE	0.69	0.56	0.62	0.76	0.65	0.70	0.84	0.86	0.85	0.99	0.86	**0.92**	0.50	0.10	0.17	0.62	0.49	0.55
WORK	0.00	0.00	0.00	1.00	0.06	0.12	0.29	0.31	0.30	1.00	0.50	**0.67**	0.08	0.25	0.13	0.47	0.50	0.48
DEMO	0.86	0.82	0.84	0.86	0.88	0.87	0.95	0.94	0.94	0.98	0.98	**0.98**	0.54	0.36	0.43	0.46	0.95	0.62
EVENT	0.71	0.57	0.63	0.86	0.29	0.43	0.73	0.76	0.74	1.00	0.81	**0.89**	0.17	0.10	0.12	0.19	0.48	0.27

(c) srpELTeC Sample (Literature)

Class	SrpNER			CNN			Jerteh			Tesla			Nano			Mini		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PERS	0.65	0.73	0.69	0.72	0.40	0.52	0.65	0.72	0.69	0.82	0.91	**0.86**	0.27	0.87	0.41	0.62	0.91	0.74
LOC	0.87	0.70	0.78	0.63	0.60	0.61	0.73	0.87	0.79	0.81	0.95	**0.88**	0.29	0.81	0.43	0.64	0.97	0.77
ORG	0.00	0.00	0.00	0.30	0.75	0.43	0.25	0.25	0.25	1.00	0.50	**0.67**	0.00	0.00	0.00	0.20	0.25	0.22
ROLE	0.77	0.83	0.80	0.61	0.20	0.30	0.88	0.39	0.54	0.96	0.54	**0.69**	0.83	0.21	0.34	0.51	0.73	0.60
WORK	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.60	0.60	1.00	0.40	**0.57**	0.10	0.40	0.16	0.36	0.80	0.50
DEMO	1.00	0.80	0.89	0.67	0.53	0.59	1.00	0.73	0.85	1.00	0.80	**0.89**	0.06	0.03	0.04	0.52	0.83	0.64

per articles, history textbook excerpts, and a literary sample. Evaluated models included a rule-based system (`SrpNER`), a `CNN`, fine-tuned transformers (`Jerteh`, `Tesla`), and prompt-based LLMs (`ChatGPT 4.o Nano` and `ChatGPT 4.1 Mini`). Fine-tuned BERT based models generally achieved the highest performance, demonstrating strong generalisation from a diverse training corpus, with `Tesla` showing exceptionally high results on the history data. The rule-based SrpNER proved robust, performing well on news and showing resilience to stylistic shifts in literary texts. The CNN was more susceptible to domain variations. Prompt-based LLMs exhibited lower performance for precise NER, particularly on the challenging literary dataset, suggesting limitations of prompting alone for complex sequence labelling tasks despite using human annotation guidelines. This analysis highlights the critical influence of both model architecture and target domain characteristics on NER performance in Serbian.

Based on these findings, our future research will focus on enhancing LLM performance for Serbian NER through refined prompting strategies (e.g., few-shot, PEFT) and exploring their potential in hybrid systems. Addressing the challenges of domain and style shifts, notably for historical/literary texts, is also crucial, potentially via dedicated domain adaptation techniques or advanced hybrid approaches. Further evaluation on a broader spectrum of Serbian text types, including lower-resource domains, is warranted. Finally, conducting detailed qualitative error analysis and exploring few-shot learning paradigms within supervised frameworks are valuable avenues for improving NER performance and reducing annotation effort in new domains.

Acknowledgments

This work was supported by the Science Fund of the Republic of Serbia (#7276, Text Embeddings – Serbian Language Applications, TESLA) and partially by the CIDEXG/2023/12 project, funded by the Generalitat Valenciana.

References

Manel Affi and Chiraz Latiri. 2022. Arabic named entity recognition using variant deep neural network architectures and combinatorial feature embedding based on cnn, lstm and bert. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 302–312.

Tom B. Brown, Benjamin Mann, Nick Ryder, and et al. 2020. Language models are few-shot learners.

Can Çetindağ, Berkay Yazıcıoğlu, and Aykut Koç. 2023. Named-entity recognition in turkish legal texts. *Natural Language Engineering*, 29(3):615–642.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the ACL: ACL-IJCNLP 2021*, pages 1835–1845, Online. ACL.

Andrija Cvejić. 2022. Prepoznavanje imenovanih entiteta u sprskom jeziku pomoću transformer arhitekture. *Zbornik radova Fakulteta tehničkih nauka u Novom Sadu*, 37(02):310–315.

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.

Maud Ehrmann, Matteo Romanello, and Amir Zeldes. 2023. Named entity recognition and classification in historical document collections. *Natural Language Engineering*.

Xin Feng, Yingrui Li, Zhang Hang, Zhang Fan, Qiong Yu, and Ruihao Xin. 2022. Tbr-ner: Research on covid-19 text information extraction based on joint learning of topic recognition and named entity recognition. *Journal of Sensors*, 2022(1):3967171.

Francesca Frontini, Carmen Brando, Joanna Byszuk, Ioana Galleron, Diana Santos, and Ranka Stanković. 2020. Named entity recognition for distant reading in ELTeC. In *CLARIN Annual Conference 2020*.

Eduardo C Garrido-Merchan, Roberto Gozalo-Brizuela, and Santiago Gonzalez-Carvajal. 2023. Comparing bert against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering*, 2(4):352–356.

Milica Ikonić Nešić, Sasša Petalinkar, Stanković Ranka, and Škorić Mihailo. 2024. BERT downstream task analysis: Named Entity Recognition in Serbian. In *14th International Conference on Information Society and Technology – ICIST 2024*. unpublished.

Milica Ikonić Nešić and Miloš Utvić. 2024. Overview of the Tesla-Ner-Nel-Gold Dataset: Showcase on Serbian-English Parallel Corpus. *Technical editors*, page 57.

Asim Jehangir, Muhammad Asad Aslam, et al. 2023. A comprehensive survey on named entity recognition: Recent advances and challenges. *Artificial Intelligence Review*.

Achraf Keraghel, Amine Abdaoui, and Abdelghani Bouramoul. 2024. Recent advances in named entity recognition: A survey. *Information Processing & Management*.

Cvetana Krstev. 2008. *Processing of Serbian. Automata, Texts and Electronic Dictionaries*. Faculty of Philology of the University of Belgrade.

Cvetana Krstev, Ivan Obradović, Miloš Utvić, and Duško Vitas. 2014. A system for named entity recognition based on local grammars. *Journal of Logic and Computation*, 24(2):473–489.

Cvetana Krstev, Ranka Stanković, Aleksandra M. Marković, and Teodora Sofija Mihajlov. 2024. Towards the semantic annotation of SR-ELEXIS corpus: Insights into Multiword Expressions and Named Entities. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 106–114, Torino, Italia. ELRA and ICCL.

Biljana Lazic and Mihailo Škoric. 2019. From DELA based dictionary to Leximirka lexical database. *Infotheca–Journal for Digital Humanities*, 19(2):00–00. https://doi.org/10.18485/infotheca.2019.19.2.4.

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

Yanan Li, Wenjie Li, Bing Qin, and Ting Liu. 2020. A survey of deep learning approaches for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Andy T. Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. 2022. Qaner: Prompting question answering models for few-shot named entity recognition.

Nikola Ljubešić and Davor Lauc. 2021. Berti\'c–the transformer language model for bosnian, croatian, montenegrin and serbian. *arXiv preprint arXiv:2104.09243*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. A. Moreiro. 2013. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Australasian Language Technology Association Workshop*, pages 51–58. Australasian Language Technology Association.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Ivana Sandrih, Cvetana Krstev, and Duško Vitas. 2019. A hybrid method for serbian personal name recognition in historical literary texts. In *RANLP*.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Computational Linguistics*, 40(2):469–510.

Hemlata Shelar, Gagandeep Kaur, Neha Heda, and Poorva Agrawal. 2020. Named entity recognition approaches and their comparison for custom ner model. *Science & Technology Libraries*, 39(3):324–337.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. Promptner: Prompt locating and typing for named entity recognition.

Ranka Stanković, Cvetana Krstev, Biljana Lazić, and Mihailo Škorić. 2018. Electronic dictionaries–from file system to lemon based lexical database. In *6th Workshop on Linked Data in Linguistic (LDL-2018), Towards Linguistic Data Science*.

Ranka Stanković, Cvetana Krstev, and Duško Vitas. 2024. SrpELTeC: A Serbian Literary Corpus for Distant Reading. *Primerjalna književnost*, 47(2).

Ranka Stanković, Cvetana Krstev, Duško Vitas, Nikola Vulović, and Olivera Kitanović. 2017. Keyword-based search on bilingual digital libraries. In *Semantic Keyword-Based Search on Structured Data Sources: COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania*, pages 112–123. Springer.

Danang Arbian Sulistyo, Didik Dwi Prasetya, Fadhli Almu'iini Ahda, and Aji Prasetya Wibawa. 2025. Pivoted low resource multilingual translation with ner optimization. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(5):1–16.

Branislava Šandrih Todorović, Cvetana Krstev, Ranka Stanković, and Milica Ikonić Nešić. 2021. Serbian NER& Beyond: The Archaic and the Modern Intertwinned. In *Deep Learning Natural Language Processing Methods and Applications – Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1252–1260.

Jovana Todorović, Cvetana Krstev, and Duško Vitas. 2021. Ner in serbian novels from the 19th century using contextual embeddings. In *INFOTEH*.

Devika Verma, Ramprasad S. Joshi, Aiman A. Shivani, and Rohan D. Gupta. 2023. Kāraka-based answer retrieval for question answering in Indic languages. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1216–1224, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Duško Vitas and Cvetana Krstev. 2012. Processing of Corpora of Serbian Using Electronic Dictionaries. *Prace Filologiczne*, 63:279–292.

Duško Vitas and Gordana Pavlović-Lažetić. 2008. Resources and methods for named entity recognition in serbian. *Infotheca*.

Duško Vitas, Ranka Stanković, and Cvetana Krstev. 2024. The Many Faces of SrpKor. In *South Slavic Languages in the Digital Environment JuDig Book of Abstracts*, Belgrade, Serbia. University of Belgrade - Faculty of Philology.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Arif Warto, Septi Handayani, et al. 2024. A systematic literature review on named entity recognition research (2011–2020). *Journal of Theoretical and Applied Information Technology*.

Yuzhe Zhang and Hong Zhang. 2023. Finbert–mrc: financial named entity recognition using bert under the machine reading comprehension paradigm. *Neural Processing Letters*, 55(6):7393–7413.

Mihailo Škorić. 2024. New Language Models for Serbian. *Infotheca – Journal for Digital Humanities*, 24(1). `https://doi.org/10.18485/2024.24.1.1`.

Mihailo Škorić and Saša Petalinkar. 2024. New XLM-R-based Language Models for Serbian and Serbo-Croatian. In *Artificial Intelligence Conference*, Belgrade. SASA.

Marko Živković, Milan Samardžić, and Ranka Stanković. 2022. Clinical named entity recognition in serbian using bert and ensemble learning. *Biomedical Signal Processing and Control*, 72.

Enhancing the Performance of Spoiler Review Detection by a LLM with Hints

Genta Nishi
Graduate School of Information
Science and Electrical Engineering,
Kyushu University, Japan
nishi.genta.985@s.kyushu-u.ac.jp

Einoshin Suzuki
Faculty of Information
Science and Electrical Engineering,
Kyushu University, Japan
suzuki@inf.kyushu-u.ac.jp

Abstract

We investigate the effects of three hints including an introduction text, a few examples, and prompting techniques to enhance the performance of a Large-Language Model (LLM) in detecting a spoiler review of a movie. Detecting a spoiler review of a movie represents an important Natural Language Processing (NLP) task which resists the Deep Learning (DL) approach due to its highly subjective nature and scarcity in data. The highly subjective nature is also the main reason of the poor performance of LLMs-based methods, which explains their scarcity for the target problem. We address this problem by providing the LLM with an introduction text of the movie and a few reviews with their class labels as well as equipping it with prompts that select and exploit spoiler types with reasoning. Experiments using 400 manually labeled reviews and about 3200 LLM-labeled reviews show that our CAST (Clue And Select Types prompting) outperforms (0.05 higher) or is on par with (only 0.01 lower) cutting-edge LLM-based methods in three out of four movies in ROC-AUC. We believe our study represents an evidence of a target problem in which the knowledge intensive approach outperforms the learning-based approach.

1 Introduction

According to the Oxford Learner's Dictionaries, a spoiler is defined as "information that you are given about what is going to happen in a film, television series, etc. before it is shown to the public"[1], which can hinder or stop consumers' enjoyment of a work (Tsang and Yan, 2009). In this paper, we focus our attention to spoiler reviews of a movie due to their complex nature for NLP and their high influence on our daily life. Manually setting mute words (Golbeck, 2012) , e.g., the true criminal, or spoiler tags, though effective, are expensive due to the necessary human labor. NLP-based automatic detection could be a realistic solution depending on its accuracy and cost.

Since movies are rich in variety and so are their reviews, detecting a spoiler review is a highly subjective task. Moreover, Guo and Ramakrishnan (2010) pointed out that constructing a large-scale dataset with high-quality labels is difficult for spoiler detection. These two reasons rule out DL-based methods from consideration, even if they have been quite successful in various NLP tasks. LLMs could be considered as the state-of-the-art solutions of the knowledge intensive approach due to their high capabilities in various tasks and their low costs in development. However, Zhang et al. (2025b) pointed out that their text classification capabilities are limited and the development has been slow, which we believe the reason for their scarcity in the spoiler detection domain.

In this paper, we investigate three kinds of hints to enhance the performance of spoiler review detection by an LLM. The first hint is an introduction text, which corresponds to domain knowledge in the knowledge intensive approach. The second hint is a few reviews with their binary class labels, i.e., spoiler or not spoiler, which can be regarded as examples for few-shot learning. The third hint is spoiler types with a reasoning strategy, which could be viewed as an inference strategy on sub-classes for the LLM. Broadly speaking, exploiting these three kinds of hints belongs to the widely-used prompt engineering, though our motivation is to obtain an evidence which suggests in the long run the characteristics and the conditions of the target problems in which the knowledge intensive approach outperforms the learning-based approach.

Figure 1 shows two working examples of our CAST on the movie "Hulk", in which Bruce transforms himself to a green heroic monster. The first

[1] https://www.oxfordlearnersdictionaries.com/definition/english/spoiler?q=spoiler

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 97–112, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-011

SPOILER
input: I thought endowing Banner's father with the Absorbing Man's powers was a brilliant idea, symbolizing what his father indirectly did to Bruce his whole life.
clue: "Absorbing Man's powers", "father indirectly did to Bruce his whole life".
spoiler type: true identity, character features, development of the story, past, problem occurs.

→ **spoiler level**: 0.9999

NOT SPOILER
input: Seeing the green behometh smash up tanks, helicopters etc had me in aweof the amazing folks who created the cgi.
clue: "seeing the green behometh", "amazing folks".
spoiler types: appearance, development of the story, true identity, past, status/power.

→ **spoiler level**: 0.1559

Figure 1: Examples of spoiler and non-spoiler reviews for "Hulk". Spoiler levels are provided by our CAST. The spoiler review mentions the identity of the final villain. The non-spoiler review mentions unimportant details.

review reveals the identity of the final villain, who gave the power to Bruce and is thus a spoiler. CAST correctly estimates its spoiler level to 0.9999 by selecting four spoiler types, of which the red two are correct, with its LLM. CAST also explains two clues in its decision, which demonstrates its comprehensibility to the users. The second review, on the other hand, just explains the widely-known capabilities of Hulk, and is thus not a spoiler. CAST correctly estimates its spoiler level again.

2 Related Work

2.1 Spoiler Detection

Spoiler Detection methods can be classified into classification-based, clues-based, and LLMs-based. The first approach exploits the powerful capabilities of the text classification methods. Wan et al. (2019) proposed SpoilerNet, which inputs review documents and item specificity information to Hierarchical Attention Network (Yang et al., 2016). Chang et al. (2021) proposed SDGNN, which combines a Graph Neural Network (Marcheggiani and Titov, 2017) that recognizes sentence dependencies with a genre aware structure. We consider this approach is inadequate for our target problem due to the lack of large-scale data and the variety in movies and their reviews.

The second approach uses multiple frameworks to extract features from various kinds of clues including user data, movie metadata, and reviews. The features could be passed to a Mixture of Experts for each genre (Zeng et al., 2024; Zhang et al., 2025a). This approach is relevant to our CAST, though the former doesn't use an LLM for the main purpose of spoiler detection.

The last approach is rare in spoiler detection, possibly due to the limited capability of LLMs in text classification (Zhang et al., 2025b). As we explained in the previous section, we try to enhance the performance of this approach by using three kinds of hints, which are not limited to the data source. Since LLMs have achieved notable successes in handling semantics (Schaeffer et al., 2025), we believe they are also promising for our target task.

2.2 Text Classification by LLM

Text classification using LLMs can be broadly classified into two approaches, i.e., the approach that relies on fine-tuning and the one on few-shot learning.

As an example of the former, Zhang et al. (2025b) proposed RGPT, of which fine-tuning is based on the idea of Adaptive Boosting (Freund and Schapire, 1997). Their fine-tuning is conducted in multiple rounds, each of which updates the weight distribution over the dataset based on the predictions of the weak learner induced in the round. The final prediction is based on weak learners with their model weights obtained according to the predictions.

As an example of the latter, Sun et al. (2023) proposed Clue and Reasoning Prompting (CARP). They pointed out that LLM-based methods are inferior to fine-tuned models in text classification tasks due to the lack of inference ability and token length limitations in the former. CARP encourages users to find clues such as keywords, tone, semantic relations, and references from the text before reasoning, which strengthen its reasoning ability. They succeeded in conducting few-shot learning by sampling a few examples with the k-nearest neighbor method and developed a voting method among LLMs with various outputs. CARP outperforms a powerful prompt engineering method Zero-shot-Chain-of-Thought[2] in text classification performance (Kojima et al., 2022).

In the datasets used for spoiler detection, the labels are typically collected from review sites[3]. It has been pointed out that their quality is low due to differences in spoiler standards between human labelers and their mistakes, e.g., they occasionally forget to add spoiler tags (Guo and Ramakrishnan, 2010). In other words, models trained on these datasets are likely to exhibit low accuracy. Therefore, we focus our attention to the second approach.

3 Target Problem

As we stated, our target problem is spoiler review detection of a movie. We could have formalized it as a classification problem by setting a binary class label of spoiler and not spoiler as our output. This formalization allows us to use accuracy as the evaluation measure, which is easy to understand, but necessitates a threshold that separates the two classes. Setting an appropriate threshold is possible when the misclassification costs, i.e., the cost of a false positive and that of a false negative, are known (Han et al., 2011), which is not the case for us. We therefore formalized the target problem as an estimation problem of the spoiler level of a movie review from 0 to 1, the higher being more likely to be a spoiler. As we will explain later, ROC-AUC (Han et al., 2011) is adopted as our evaluation measure.

The input to our target problem consists of set $\{R_{i,1}, \ldots, R_{i,n(i)}\}$ of review texts and introduction text I_i for movie i, where $R_{i,j}$ and $n(i)$ represent the j-th review and the number of reviews, respectively. The output of our target problem is spoiler levels $(Y_{i,1}, \ldots, Y_{i,n(i)})$, where $Y_{i,j}$ represents the spoiler level of $R_{i,j}$.

Since the reviews can be sorted in descending order based on their spoiler levels, we can compute ROC-AUC of an output, which we adopt as our evaluation measure (Han et al., 2011). We assume that the class label of $R_{i,j}$ is available in everything in the output. ROC-AUC corresponds to the probability that a positive example, i.e., a spoiler review in our case, is ranked higher than a negative example, i.e., a non spoiler. ROC-AUC is widely adopted in detection problems where the misclassification costs are unknown.

4 Proposed Method: CAST

4.1 Overview

Algorithm 1 CAST

Input: set $\{R_{i,1}, \ldots, R_{i,n(i)}\}$ of review documents and introduction text I_i of movie i.
Output: spoiler levels $(Y_{i,1}, \ldots, Y_{i,n(i)})$

```
for j = 1 to n(i) do
    for each sentence r_{i,j,k} in R_{i,j} do
        // Construct prompt_{i,j,k}.
        c_{i,j,k} = CLUE(r_{i,j,k})
        t_{i,j,k} = SelectType(r_{i,j,k}, c_{i,j,k})
        prompt_{i,j,k}
            = BasePrompt(I_i, r_{i,j,k}, c_{i,j,k}, t_{i,j,k})
        // Estimate P_ANSWER using LLM.
        P_SPOILER = P(" SP"|prompt_{i,j,k})
        P_NOT SPOILER = P(" NOT"|prompt_{i,j,k})
        // Compute the spoiler level.
        y_{i,j,k} = e^{P_SPOILER} / (e^{P_SPOILER} + e^{P_NOT SPOILER})
    end for
    Y_{i,j} = max_k y_{i,j,k}
end for
𝒴_i = (Y_{i,1}, ..., Y_{i,n(i)})
return 𝒴_i
```

$$y_{i,j,k} = \frac{e^{P_{\text{SPOILER}}}}{e^{P_{\text{SPOILER}}} + e^{P_{\text{NOT SPOILER}}}}$$

$BasePrompt(I_i, r_{i,j,k}, c_{i,j,k}, t_{i,j,k})$ is shown below.

> This is a Spoiler Detection for input movie reviews.
> "Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person's first experience of the work.
> A significant plot point is one that cannot be predicted from the film's introduction or early developments.
> List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler

[2] Kojima et al. (2022) succeeded in stabilizing the zero-shot performance of LLMs and improving performance even in the few-shot case by using the CoT (Chain-of-Thought) approach.

[3] IMDb (https://www.imdb.com/) and Goodreads (https://www.goodreads.com/)

detection of the input.
Finally, based on introduction, clues, spoiler types, and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [example introduction 1]
review: [example review 1]
clue: [example clues 1]
spoiler types: [example types 1]
answer: [example answer 1]
...(7 few-shot examples follow.)

introduction: I_i
review: $r_{i.j.k}$
clue: $c_{i,j,k}$
spoiler types: $t_{i,j,k}$
answer:

We propose CAST (Clue And Select Types prompting), a spoiler detection method using an LLM. As shown later in Figure 2, CARP is weak against roundabout expressions, which are common in spoiler detection. We attribute this reason to the fact that such expressions "confuse" the LLM's judgment. Therefore, in CAST, the LLM is dynamically given spoiler types as hints for the judgment.

First, clues $CLUE(r_{i,j,k})$ are extracted from the input review $r_{i,j,k}$ according to Sun et al. (2023), where $r_{i,j,k}$ represents the k-th sentence of $R_{i,j}$. Then, the LLM is given most of $BasePrompt$, from the beginning to the second "clue:" so that it outputs phrases that are clues for spoiler detection. We show the example reviews in Tables 10, 11, and 12. Next, the LLM is given all $BasePrompt$, which includes the output above as $CLUE(r_{i,j,k})$ and the spoiler types obtained with $SelectType(_{i,j,k}, CLUE(r_{i,j,k}))$, which we will explain in the next Sections.

Next, based on the input, clues, and spoiler types, LLM outputs a probability distribution of the following tokens: "SPOILER", "NOT SPOILER", and other words[4]. From the distribution, we calculate the probability P_{SPOILER} that the LLM outputs "SPOILER" and the probability $P_{\text{NOT SPOILER}}$ that it outputs "NOT SPOILER"[5]. Finally, we calculate the spoiler level $y_{i,j,k}$ using the softmax function to eliminate the probability of other words, i.e., the probabilities of "SP" and "NOT" sum up to 1. $Y_{i,j}$ is the maximum value of $y_{i,j,k}$ in terms of k, as we think the sentence that is most likely to be a spoiler determines the spoiler level of the review.

To provide diverse input for the LLM, we used eight few-shot examples that covered a range of review types (a direct spoiler review, an indirect spoiler review, an impression-only review, and a review with unimportant content). In addition, these examples were drawn from movies across various genres to develop a method applicable to multiple domains.

character relationships	true identity
character features	life or death
victory or defeat	purpose
problem occurs	trick
development of the story	past
status/power	appearance

Table 1: Spoiler types defined by Tajima and Nakamura (2015).

4.2 Selecting Spoiler Types

Since only one or a few spoiler types are relevant to a review, inputing all 12 types to the LLM would degrade the performance. We thus propose to select relevant spoiler types using the LLM using the following prompt.

Please select k spoiler types that are most appropriate for the review and its keywords from the following spoiler types.

spoiler type: [all types]
review: [review]
keywords: [clues]
appropriate type:

In CAST, we use the spoiler types classified by Tajima and Nakamura (2015). They collected 1370 spoilers from over 100 students and manually classified them into 12 types without any excess or deficiency. We show them in Table 1.

[4] We can obtain the distribution by setting the variable *logprobs* to True in llama-cpp-python (`https://github.com/abetlen/llama-cpp-python`).

[5] To be precise, we use "SP" and "NOT" instead of "SPOILER" and "NOT SPOILER", respectively, as the last two are not included in the vocabulary of Llama. These replacements are justified because the probabilities of "ILER" and "SPOILER" right after "SP" and "NOT" are almost 1, respectively.

5 Experiments

5.1 Conditions

As datasets, Kaggle (Misra, 2022) and LCS (Wang et al., 2023) are often used in recent spoiler detection studies (Zeng et al., 2024; Zhang et al., 2025a). However, several papers argue that their labels are not accurate due to their human labelers, whose spoiler standards are not uniform (Guo and Ramakrishnan, 2010; Wan et al., 2019). We conduct experiments on the IMDb dataset (Misra, 2022), which one annotator relabeled manually[6]. We also conducted the relabeling with an LLM. In our relabeling, we define a spoiler review as a review that includes an important event shown in Table 7 in the Appendix. The target movies and data sizes that we used in our experiments are shown in Table 2. The introduction texts were taken from the IMDb movie page. We show them in Table 9.

In the relabeling with an LLM, we adopted the following prompt.

> This is to determine whether a review contains spoilers.
> "Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person's first experience of the work.
> A significant plot point is one that cannot be predicted from the film's introduction or early developments.
> We will give you the title and significant plots of the movie, so please use that to determine whether the review contains spoilers.
>
> title: [title]
> significant plots: [events]
> review: [review]
> label (True or False):

Here, [events] is the same as the event shown in Table 7. In the relabeling, we used Llama3.1-8B (Dubey et al., 2024). Table 3 shows the ratios of the modified labels in our relabeling.

5.2 Baseline Methods

We employed Zero-Plus-Few-shot-Chain-of-Thought (CoT) (Kojima et al., 2022) and CARP (Sun et al., 2023) as the baseline methods. Although CoT is not a method developed for text classification, we use it as a baseline following Sun et al. (Sun et al., 2023). As we have introduced, CARP is a method for text classification by LLM. To keep the setting fair, we did not employ its voting method. We also tested variants of these methods by omitting their reasoning process and/or by employing the introduction text. Prompts of the methods are shown in the Appendix. In comparing CAST with the baseline methods, we used Llama2-13B (Touvron et al., 2023) implemented in llama-cpp-python[7] as the backbone of the LLM. In this experiment, we used Human labels. CAST and CARP were also compared in experiments using LLM labels with Llama2-13B, as well as in experiments using human labels on more recent LLM platform, Llama3.1-8B (Dubey et al., 2024). We adopted +i-r as the condition due to its overall, superior performance in the latter.

5.3 Few-shot Learning

Few-shot leaning is performed to standardize the answer format and improve accuracy. Two reviews (positive and negative) were collected from each of four movies ("Million Dollar Baby", "The Fast and the Furious", "Groundhog Day", and "Match Point") in the IMDb dataset (Misra, 2022). To be fair, the same examples were used by all methods[8]. The examples are shown in Tables 10, 11 and 12.

5.4 Results

The results are shown in Table 4. We first focus on the results on the datasets relabeled by a human, which are considered more accurate those with the LLM. For "Hulk" and "The Shawshank Redemption", CAST is the best method. For "Mean Girls", it is the third best performing method, quite close to the second one. For "Blood Diamond", it is the second best performing method overall and the best performing method for "+intro -reasoning". Overall, we conclude that CAST is the best method for few-shot spoiler detection based on human values. We then focus on the results on the LLM relabeled dataset. Compared to CARP, it performs worse on "Hulk" but slightly better on the other three movies. A detailed analysis is provided in Section 6.

[6]We admit the weakness of adopting a single annotator as the quality is affected by his subjectivity.

[7]abetlen/llama-cpp-python. `https://github.com/abetlen/llama-cpp-python`

[8]The presence or absence of spoiler types or introduction text is adjusted to match the method. The sampling method of CARP was skipped as the examples were given.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
Human	100	100	100	100
-spoiler	31	43	32	25
-not spoiler	69	57	68	75
LLM	523	1737	445	628
-spoiler	41	148	84	117
-not spoiler	482	1589	361	511

Table 2: Target movies and data size of the dataset.

	Human	LLM
Hulk	36%	83.2%
The Shawshank Redemption	43%	49.6%
Mean Girls	33%	73.7%
Blood Diamond	25%	74.4%

Table 3: Ratios of modified labels in our relabeling.

6 Detailed Analysis

6.1 Case Study

To investigate how CAST detects spoilers, we analyze the example of "Blood Diamond" in comparison with CARP. The following contains spoilers for "Blood Diamond". In the final scene, the main character (Leonardo DiCaprio) dies. This content is clearly a spoiler. The review of Figure 2 includes this content, but describes it in a roundabout way ("I was hoping Leo would not die"). CARP is affected by the roundabout expression and shows poor performance, i.e., the spoiler level of about 0.78 is moderately high. On the other hand, in CAST, we can see that the spoiler types, e.g., "life or death", selected dynamically lead the LLM to output a very high spoiler level of about 0.97.

CARP
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn't about happy endings which i admire.
clue: "Leo", "die", "Zwick isn't about happy endings".

→ **spoiler level**: 0.7886 △

CAST
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn't about happy endings which i admire.
clue: "Leo", "die", "Africa".
spoiler types: life or death, true identity, development of the story, victory or defeat, problem occurs.

→ **spoiler level**: 0.9729 ◯

Figure 2: Example of spoiler detection in a review of "Blood Diamond".

6.2 Useful Issues

We present several issues that we noticed in the experiments, which could contribute to our future research.

6.2.1 LLM is sensitive to cruel scenes

LLM over-identifies scenes involving injury or death as spoilers. Usually, commenting on an injury of a sub-character, especially toward the beginning of the movie, is not a spoiler. This movie, "Blood Diamond", is set in a war zone and thus contains many violent scenes, which are related to the spoiler type "life or death". Not only CAST but also CARP and CoT are subject to this kind of false positives, as they all employ LLMs. A

		Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
			Human Relabel		
Llama2-13B	CoT	0.7447	0.7209	0.8162	<u>0.7072</u>
	+i	0.7176	0,7187	0.8580	**0.7211**
	-r	0.6397	0.7340	0.7762	0.6450
	+i -r	0.7218	0.7546	**0.8736**	0.5691
	CARP	0.7433	0.7623	0.8350	0.6651
	-f	0.6840	**0.8209**	0.6719	0.6347
	+i	0.7087	0.7325	0.8244	0.6705
	-r	0.7555	0.7475	0.7992	0.6373
	+i -r	0.7129	<u>0.7823</u>	0.8534	0.5968
	CAST	<u>0.7685</u>	0.7638	0.8208	0.7056
	-f	0.7162	0.7825	0.7849	0.5696
	+i	**0.7761**	0.7813	<u>0.8603</u>	0.6863
Llama3.1-8B	CARP +i -r	0.7232	0.7987	0.8470	0.7109
	CAST +i	**0.7377**	**0.8184**	**0.8732**	**0.7701**
			LLM Relabel		
Llama2-13B	CARP +i -r	**0.8505**	0.8149	0.8272	0.7283
	CAST +i	0.8219	**0.8467**	**0.8273**	**0.7395**

Table 4: ROC-AUC of the spoiler levels of each methods for four movies from the IMDb dataset (Misra, 2022). "-f" represents a case that the prompt contains no few-shot example. "+i" represents a case that the prompt contains an introduction of the movie. "-r" represents a case without reasoning, which corresponds to our CAST. "+r" represents a case with reasoning, of which details will be explained in Section 7.2. The highest value for each film is highlighted in bold fonts, the second highest in underlined.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
AllType	0.7602	0.7772	0.8695	0.6768
Embedding	0.7662	0.7764	0.8355	0.6645
LLM($k = 1$)	0.7017	0.7919	0.8125	0.6864
LLM($k = 3$)	0.7639	0.7597	0.8566	0.6704
LLM($k = 5$)	0.7761	0.7813	0.8603	0.6863

Table 5: ROC-AUC for each spoiler type selection method.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
CARP +reasoning	0.7087	0.7325	0.8244	0.6705
CARP -reasoning	0.7120	0.7746	0.8621	0.6277
CAST +reasoning	0.7139	0.7195	0.8235	0.7072
CAST -reasoning	0.7761	0.7813	0.8603	0.6863

Table 6: ROC-AUC for each method with and without reasoning.

input: The rebels make a speech and then cut some kids arm off, then there ready to do the same to Solomon, but the rebel leader decides to spare him and take him as a prisoner and use him as a worker, then the movie continues on from there.
clue: "cut some kids arm off", "spare him and take him as a prisoner".
spoiler type: life or death, problem occurs, development of the story, past, status/power.

→ **spoiler level**: 0.9997

Figure 3: Example which shows LLM is sensitive to cruel scenes.

possible solution would be to strengthen the movie introduction to discourage LLM from reacting to the early scenes, or to use the synopsis included in the IMDb dataset (Misra, 2022) to convince LLM that the scenes are not important.

6.2.2 Spoiler type can increase false positive

Although spoiler types provide evidence of spoilers and contribute to lowering the false negative rate (Figure 2), they may also help to judge a non spoiler example as a spoiler. We show an example in Figure 4, which includes known descriptions of the main character in "Hulk". Unlike CARP (-reasoning) which appropriately gives a low spoiler level (0.1556), our CAST gave a high spoiler level due to the spoiler types "past" and "character features". Possible solutions include explaining in the prompt that some reviews may not be spoilers even if they match the spoiler type, or setting types also to the class of not spoiler. We suspect that there are about three such types but believe we need more evidence for further investigation.

input: Young Bruce grows up in an adopted family, never knowing what happened to his birth parents nor that he may be carrying abnormal genes as a result of his father's work.
clue: "Young Bruce", "never knowing", "abnormal genes".
spoiler type: past, character features, true identity, development of the story, problem occurs.

→ **spoiler level**: 0.9430

Figure 4: Example which shows spoiler types lead to an excessive spoiler level.

7 Ablation Study

7.1 Methods for Selecting Types

We evaluate the effect of our spoiler type selection in Section 4.1, which we call LLM. Here, we set the number of choices $k = 1, 3, 5$ and use Llama2 (Touvron et al., 2023) as our LLM.

As an alternative, we introduce another method which we call Embedding. Following the Dense Passage Retriever (DPR) (Karpukhin et al., 2020), we selected the spoiler type by the cosine similarity between the embedding vectors of the clues and the spoiler type. The embedding model is RoBERTa (Liu et al., 2019), which is fine-tuned on the spoiler domain dataset (Wan et al., 2019) and its paraphrases by Llama2-13B. The loss function is the same as DPR.

As a baseline method, we also compare AllType that does not select spoiler types and uses all of them. The experimental settings are based on Section 5. We use the human-relabeled datasets. The results are shown in Table 5. Overall, we conclude that the results of LLM ($k = 5$) is the best. Furthermore, an example is shown in Figure 5. This example is about "the death of a person", but Embedding cannot select "life or death", resulting in a false negative. In contrast, LLM is able to cor-

Embedding
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn' t about happy endings which i admire.
clue: "Leo", "die", "Africa".
spoiler type: problem occurs, past, trick, character features, appearance, character relationships.

→ **spoiler level**: 0.5457

LLM ($k = 5$)
...
spoiler type: life or death, true identity, development of the story, victory or defeat, problem occurs.

→ **spoiler level**: 0.9729

Figure 5: Comparison of Select type methods: Embedding and LLM ($k = 5$).

rectly select the spoiler type, and the output is also correct. We used in the main experiments LLM ($k = 5$) as our selection method.

7.2 Effect of Reasoning

Though reasoning is said to enhance the performance of LLM (Wei et al., 2022; Kojima et al., 2022; Sun et al., 2023), several researchers argue against it. Chen et al. (2024) point out several cases in which reasoning increases the probability of an incorrect output in text classification. Therefore, we investigate the effect of reasoning in CARP and CAST. We write "+reasoning" and "-reasoning" for with and without reasoning, respectively. We give their prompts at the end of the Appendix. The experimental settings are based on Section 5. We use the human-relabeled datasets.

The results are shown in Table 6. In both CARP and CAST, "-reasoning" performs better. In fact, there are almost no case where the correct answer is obtained through reasoning. We conclude that reasoning is unnecessary for our spoiler detection.

8 Conclusion

We show that in the field of spoiler detection, where there is a lack of high-quality datasets, adding three kinds of hints improves the performance of LLM-based spoiler review detection of a movie. It is no wonder that the introduction text and the few examples for few-shot learning are useful, as they represent typical domain knowledge and representative cases. The types that we used represent subclasses of the positive class. Our prompts instructs their effective selection and usage, which could be also useful in other text classification problems.

Our future research includes defining better spoiler types as well as setting non-spoiler types. Such types or sub-classes could be set dynamically according to the given reviews, the introduction text, and the few examples for few-shot learning. Utilizing other kinds of additional data such as synopses would deepen our understanding on the target domain and the prompt engineering.

References

Buru Chang, Inggeol Lee, Hyunjae Kim, and Jaewoo Kang. 2021. "Killing Me" Is Not a Spoiler: Spoiler Detection Model using Graph Neural Networks with Dependency Relation-Aware Attention Mechanism. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3613–3617.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Jennifer Golbeck. 2012. The Twitter Mute Button: a Web Filtering Challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2755–2758.

Sheng Guo and Naren Ramakrishnan. 2010. Finding the Storyteller: Automatic Spoiler Tagging using Linguistic Cues. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 412–420.

Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques, 3rd ed.* Morgan Kaufmann.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-Tau Yih. 2020. Dense Passage Retrieval for

Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6769–6781.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

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: 2403.05265.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. arXiv: 1703.04826.

Rishabh Misra. 2022. IMDb Spoiler Dataset. arXiv: 2212.06034.

Rylan Schaeffer, Punit Singh Koura, Binh Tang, Ranjan Subramanian, Aaditya K Singh, Todor Mihaylov, Prajjwal Bhargava, Lovish Madaan, Niladri S Chatterji, Vedanuj Goswami, et al. 2025. Correlating and Predicting Human Evaluations of Language Models from Natural Language Processing Benchmarks. arXiv: 2502.18339.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text Classification via Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Kazuki Tajima and Satoshi Nakamura. 2015. A Study on Story Spoilers and Considering the Possibility to Detect Spoilers. *IPSJ SIG Technical Report on Groupware and Network Services (GN)*, 2015(7):1–6. (in Japanese).

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: 2307.09288.

Alex S. L. Tsang and Dengfeng Yan. 2009. Reducing the spoiler effect in experiential consumption. pages 708–709.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2605–2610.

Heng Wang, Wenqian Zhang, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. Detecting Spoilers in Movie Reviews with External Movie Knowledge and User Networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16035–16050.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Zinan Zeng, Sen Ye, Zijian Cai, Heng Wang, Yuhan Liu, Haokai Zhang, and Minnan Luo. 2024. MMoE: Robust Spoiler Detection with Multi-modal Information and Domain-aware Mixture-of-Experts. arXiv: 2403.05265.

Haokai Zhang, Shengtao Zhang, Zijian Cai, Heng Wang, Ruixuan Zhu, Zinan Zeng, and Minnan Luo. 2025a. Unveiling the Hidden: Movie Genre and User Bias in Spoiler Detection. arXiv: 2504.17834.

Yazhou Zhang, Mengyao Wang, Qiuchi Li, Prayag Tiwari, and Jing Qin. 2025b. Pushing the limit of LLM capacity for text classification. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1524–1528.

A Details of the Experiments

We show important events of each movie in Table 7. Each sentence is taken from plot_synopsis of IMDb_movie_details in the IMDb Dataset (Misra, 2022). We show the prompts of baseline methods in Table 8. We show the introduction text of each movie in Table 9. Each text is taken from the movie's IMDb page.

We show the reviews, the clues, the spoiler types, the reasonings, and the answers used in the few-shot learning in Tables 10 and 11.

We also show the introduction texts in the few-shot learning in Table 12.

We show the prompts with and without reasoning for CARP and CAST.

CARP

prompt of + reasoning

> This is a Spoiler Detection for input movie reviews.
> List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.
> Next, deduce the diagnostic REA-

Hulk	There he proceeds to wreak havoc in the city until Betty arrives and calms him down. David taps into a powerline and becomes living electricity. Bruce transforms into the Hulk and the two men battle.
The Shawshank Redemption	Red believes Andy intends to use the hammer to engineer his escape in the future but when the tool arrives and he sees how small it is, Red puts aside the thought that Andy could ever use it to dig his way out of prison. He goes to a halfway house but finds it impossible to adjust to life outside the prison. He eventually commits suicide.
Mean Girls	In her efforts to get revenge on Regina, Cady gradually loses her individual personality and remakes herself in the image of Regina. She soon becomes as spiteful as Regina, abandoning Janis and Damien and focusing more on her image. Regina storms out, pursued by an apologetic Cady, and gets hit by a school bus in her haste. At the Spring Fling dance, Cady is elected Spring Fling Queen, but in her acceptance speech, she declares her victory is meaningless: they are all wonderful in their own way and thus the victory belongs to everyone.
Blood Diamond	Dia is conscripted into the rebel forces, the brainwashing eventually turning him into a hardened killer. Archer holds off the soldiers chasing them while Solomon and Dia flee, and then makes a final phone call to Bowen, asking her to help Solomon as a last favor before looking out over the beautiful landscape of Africa once more and dying peacefully.

Table 7: Important events of each movie.

CoT (Kojima et al., 2022)	
	You are detecting “Spoilers” in movie reviews. “Spoilers” is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person’s first experience of the work. A significant plot point is one that cannot be predicted from the filmś introduction or early developments. Based on introduction, does the following review contain spoilers? introduction: [intro] review: [review] reasoning: Let’s think step by step. [reasoning]
CARP (Sun et al., 2023)	
	This is a Spoiler Detection for input movie reviews. List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input. Next, deduce the diagnostic REASONING process from premises (i.e., introduction, clues, input) that support the spoiler detection. Finally, based on the introduction, the clues, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER. introduction: [intro] review: [review] clues: [clue] reasoning: [reasoning]

Table 8: Prompts of the baseline methods.

Hulk	Bruce Banner, a genetics researcher with a tragic past, suffers a lab accident that makes him transform into a raging, giant green monster when angered, making him a target of forces seeking to abuse his power.
The Shawshank Redemption	A banker convicted of uxoricide forms a friendship over a quarter century with a hardened convict, while maintaining his innocence and trying to remain hopeful through simple compassion.
Mean Girls	Cady Heron is a hit with The Plastics, the A-list girl clique at her new school, until she makes the mistake of falling for Aaron Samuels, the ex-boyfriend of alpha Plastic Regina George.
Blood Diamond	A fisherman, a smuggler, and a syndicate of businessmen match wits over the possession of a priceless diamond.

Table 9: Introduction texts of the movies.

Million Dollar Baby	input: I can't find any reason for not loving this movie as much as possible. clues: "I can't find", "loving". spoiler type: life or death, character features, development of the story, appearance. reasoning: This review is just an opinion like "can't find" and "loving" and does not touch on the content of the movie. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: When Maggie finally gets her title fight, an illegal punch by her monster-like opponent sends her to the mat, landing head-first on her corner stool- an event which in real life would disqualify her opponent and possibly concuss Maggie instead wins her opponent the fight and renders Maggie paralyzed, bedridden and ventilator-dependent for the rest of her miserable life. clues: "an illegal punch", "Maggie paralyzed, bedridden", "the rest of her miserable life". spoiler type: problem occurs, life or death, character features, past, character relationships, development of the story, appearance. reasoning: This review is about Maggie suffering a concussion and becoming bedridden, an event that changes her life and is a key plot of the movie. Therefore, this review matches "problem occurs" and "development of the story". answer: SPOILER.
The Fast and the Furious	input: It is plot is plain and predictable but because it's unique itself and is origin of all illegal street racing movies so this makes the meaning of the plot inconsequential. clues: "plot is plain and predictable". spoiler type: past, trick, appearance. reasoning: This review criticizes the storyline but does not reveal any specifics. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: It's beautiful to see Brian and Dom at the end: Brian betrayed him and should arrest him but instead, they do the 10 second-race and don't know what to think about each other. clues: "It's beautiful to see Brian and Dom", "Brian betrayed him", "should arrest him", "the 10 second-race". spoiler type: life or death, true identity, character features, trick, past, character relationships, appearance, development of the story. reasoning: This review is about the last scene of the movie and my thoughts on that scene. Although it contains thoughts, this review contains the important plot of the last scene of the movie. Therefore, this review matches "character relationships" and "development of the story". answer: SPOILER.

Table 10: Reviews, the clues, spoiler types, the reasonings, and the answers used in few-shot learning, part 1. The information to be used is determined according to the conditions of each method (e.g., "reasoning" is omitted for methods that do not perform reasoning).

Groundhog Day	input: This pattern happens over and over again until he realizes he cannot escape Groundhog Day. clues: “happens over and over again”, “cannot escape Groundhog Day”. spoiler type: life or death, character relationships, character features, appearance. reasoning: This review talks about the film repeating the same day over and over again, which is the premise of the film and is also used in the film’s introduction. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: As, most notably, is the way that Andie MacDowell’s Rita can so magically change her opinion of Phil the second that she finds out that he plays an instrument. clues: “Rita”, “magically change her opinion”. soiler type: problem occurs, character relationship, life or death, character features, trick, appearance. reasoning: The review notes that Rita eventually develops feelings for Connors, which is a key plot point in the film’s final conclusion. Therefore, this review matches “tricks” and “character relationship”. answer: SPOILER.
Match Point	input: Sensing an opportunity to climb the social ladder he starts seeing her just as he meets Nola Rice (Scarlett Johanssen), an aspiring American actress, whom he openly flirts with until he realizes she’s Tom’s girlfriend, but an outsider in the Wilton household. clues: “starts seeing her”, “he openly flirts”. spoiler type: past, character features, trick. reasoning: The review notes that Chris begins an affair, but this is just an introduction to the film and not a major plot point. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: Then she starts to get clingy and so he kills her. clues: “starts to get clingy”, “he kills her”. spoiler type: problem occurs, life or death, true identity, character features, appearance. reasoning: The review is about a woman who becomes annoyed with a man and ends up killing her. Therefore, this review matches “life or death” and “problem occurs”. answer: SPOILER.

Table 11: Reviews, the clues, spoiler types, the reasonings, and the answers used in few-shot learning, part 2.

Million Dollar Baby	Frankie, an ill-tempered old coach, reluctantly agrees to train aspiring boxer Maggie. Impressed with her determination and talent, he helps her become the best and the two soon form a close bond.
The Fast and the Furious	Los Angeles police officer Brian O'Conner must decide where his loyalty really lies when he becomes enamored with the street racing world he has been sent undercover to end it.
Groundhog Day	A narcissistic, self-centered weatherman finds himself in a time loop on Groundhog Day.
Match Point	At a turning point in his life, a former tennis pro falls for an actress who happens to be dating his friend and soon-to-be brother-in-law.

Table 12: Introduction texts of the movies in the few-shot learning.

SONING process from premises (i.e., introduction, clues, input) that support the spoiler detection.
Finally, based on the introduction, the clues, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.
introduction: [intro]
review: [review]
clues: [clue]
reasoning: [reasoning]
answer:

prompt of – reasoning

This is a Spoiler Detection for input movie reviews.
List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.
Finally, based on the introduction, the clues and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.
introduction: [intro]
review: [review]
clues: [clue]
answer:

CAST

prompt of + reasoning

This is a Spoiler Detection for input movie reviews.
"Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil person's first experience of the work.
A significant plot point is one that cannot be predicted from the film's introduction or early developments.
List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.
Next, deduce the diagnostic REASONING process from premises (i.e., introduction, clues, input) that support the spoiler detection.
Finally, based on introduction, clues, spoiler types, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.
introduction: [intro]
review: [review]
clues: [clue]
spoiler type: [types]
reasoning: [reasoning]
answer:

prompt of - reasoning

This is a Spoiler Detection for input movie reviews.
"Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person's first experience of the work.
A significant plot point is one that cannot be predicted from the film's introduction or early developments.
List CLUES (i.e., keywords, phrases,

contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.
Finally, based on introduction, clues, spoiler types, and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.
introduction: [intro]
review: [review]
clues: [clue]
spoiler type: [types]
answer:

Evaluating Structured Decoding for Text-to-Table Generation: Evidence from Three Datasets

Julian Oestreich[1] and **Lydia Müller**[1,2]

[1]Institute for Applied Informatics (InfAI) at Leipzig University
[2]Leipzig University

Abstract

We present a comprehensive evaluation of structured decoding for text-to-table generation with large language models (LLMs). While previous work has primarily focused on unconstrained generation of tables, the impact of enforcing structural constraints during generation remains underexplored. We systematically compare schema-guided (structured) decoding to standard one-shot prompting across three diverse benchmarks - E2E, Rotowire, and Livesum - using open-source LLMs of up to 32B parameters, assessing the performance of table generation approaches in resource-constrained settings. Our experiments cover a wide range of evaluation metrics at cell, row, and table levels. Results demonstrate that structured decoding significantly enhances the validity and alignment of generated tables, particularly in scenarios demanding precise numerical alignment (Rotowire), but may degrade performance in contexts involving densely packed textual information (E2E) or extensive aggregation over lengthy texts (Livesum). We further analyze the suitability of different evaluation metrics and discuss the influence of model size.

1 Introduction

Automatically converting text into structured tables has become a key challenge in information extraction and data-driven reporting. By converting unstructured content into tables, downstream tasks such as knowledge-base construction (Liu et al., 2023; Kruit et al., 2020), document summarization, and web chatbot readability (Chen et al., 2025) can be improved. Early work framed the task as a sequence-to-sequence learning problem using encoder-decoder architectures (Wu et al., 2022), while the more recent approaches leverage LLMs with different prompting techniques, often over multiple stages. While recent advances in constrained decoding and grammar-based generation have led to improvements in structured output tasks, these methods have not yet been systematically applied to the text-to-table task. As a result, the impact of enforcing structural constraints during generation remains underexplored. Our contribution is as follows: We compare schema-guided decoding to one-shot prompting on E2E, Rotowire, and Livesum to assess how schema enforcement impacts validity and semantic quality of the resulting markdown tables at the cell, row, and table levels. We also examine model-size effects using open-source LLMs up to 32B parameters and evaluate metric suitability to guide future text-to-table research.

2 Related work

The Text-to-Table generation task was introduced by (Wu et al., 2022), who framed it as a sequence-to-sequence learning problem within the field of information extraction. They were using fine-tuned BART-based models on pairs of texts and tables to predict table representations based on the textual input. Further works have been using Large Language Models (LLMs) to solve the problem, with key differences regarding the type of generated markup sequence for table representation (Tang et al., 2024), their prompting techniques (Coyne and Dong, 2024), underlying datasets and whether the overall table format (schema) was provided to the model or not. The majority of the works was providing the table schema either while training or prompting (Coyne and Dong, 2024; Jiao et al., 2023; Tang et al., 2024), while the recent work of Ahuja et al. (2025) did not. While these prior studies have advanced text-to-table generation using either large-scale proprietary models or fine-tuned open-source LLMs—often ranging from 7B to 70B parameters or more, our work systematically investigates schema-guided decoding in smaller, publicly available open-source models. Recent advances in generative artificial intelligence have shown, that the generation of structured out-

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 113–122, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-012

puts, such as tables, could benefit from constrained decoding strategies (Park et al., 2025; Geng et al., 2023). Notably, Geng et al. (2023) found that ”grammar-constrained LMs substantially outperform unconstrained LMs” on structured NLP tasks like information extraction and constituency parsing, while Tam et al. (2024) found, that these decoding strategies degrade the semantical correctness of generated outputs. Most existing evaluations focus on general text generation or specialized NLP tasks, suggesting that the impact of constrained decoding is task-dependent. This leaves gaps in our understanding of how constraints affect content generation for structured data representations, such as tabular data.

3 Methodology

3.1 Data

Our benchmarking data consists of three datasets, each with distinct characteristics spanning a spectrum of text-to-table challenges: the E2E dataset (Novikova et al., 2017), the Rotowire dataset (Wiseman et al., 2017), and the Livesum dataset (Deng et al., 2024). E2E and Rotowire were originally designed for the Table-to-Text generation task, while Livesum was specifically created for the Text-to-Table task. E2E and Rotowire, along with WikiTableText (Bao et al., 2018) and WikiBio (Lebret et al., 2016), were repurposed for Text2Table evaluation by Wu et al. (2022). However, aside from Rotowire, these repurposed datasets lack structural diversity, as they only feature simple tables with two columns. For this reason, we only include E2E as a representative of simple tables in our experiments.

Descriptive statistics on the datasets show that while the input texts for E2E are very short, with an average of 24 words, the sizes in the other two datasets are significantly larger: Rotowire with an average of 308 words, and Livesum with 1,138 words on average (see Table 1). While E2E and Livesum show a uniform distribution of row- and column sizes, the Rotowire tables have a greater diversity regarding their sizes. E2E, sourced from the restaurant domain, consists of short textual descriptions paired with two-row tables summarizing restaurant attributes. Its focus lays on extracting textual information from short texts into simple tables. The Rotowire dataset originates in the sport domain and is a widely used benchmark in natural language generation and information extraction (Sharma et al., 2024; Puduppully et al., 2019). Each example contains one or more tables of statistics on basketball players and teams (e.g., points, assists, rebounds), paired with human-written game summaries and it requires the identification and assignment of sparsely mentioned numerical statistics to player and team tables. Livesum (Deng et al., 2024) also comes from the sports domain and comprises live soccer commentaries together with team statistics. It demands the truthful aggregation of atomic extraction units, such as individual events (e.g., goals, fouls), that are distributed throughout a longer text and organize them into comprehensive tables, a task that likely demands enhanced reasoning capabilities of the models.

Game Summary:
The Atlanta Hawks (46 - 12) beat the Orlando Magic (19 - 41) 95 - 88 on Friday. Al Horford had a good all - around game, putting up 17 points, 13 rebounds, four assists and two steals in a tough matchup against Nikola Vucevic. Kyle Korver was the lone Atlanta starter not to reach double figures in points. Jeff Teague bounced back from an illness, he scored 17 points to go along with seven assists and two steals. After a rough start to the month, the Hawks have won three straight and sit atop the Eastern Conference with a nine game lead on the second place Toronto Raptors. The Magic lost in devastating fashion to the Miami Heat in overtime Wednesday. They blew a seven point lead with 43 seconds remaining and they might have carried that with them into Friday’s contest against the Hawks. Vucevic led the Magic with 21 points and 15 rebounds. Aaron Gordon (ankle) and Evan Fournier (hip) were unable to play due to injury. The Magic have four teams between them and the eighth and final playoff spot in the Eastern Conference. The Magic will host the Charlotte Hornets on Sunday, and the Hawks will take on the Heat in Miami on Saturday.

Team:

Team	Losses	Total points	Points in 4th quarter	Wins
Hawks	12	95		46
Magic	41	88	21	19

Player:

Player	Assists	Points	Total rebounds	Steals
Nikola Vucevic		21	15	
Al Horford	4	17	13	2
Jeff Teague	7	17		2

Figure 1: Example from the Rotowire dataset showing a game summary with the corresponding team and player box scores.

3.2 Prompting & Generation

For generating structured tables from text using Large Language Models (LLMs), we follow two different methods: free-form (unstructured) generation and schema-guided (structured) decoding.

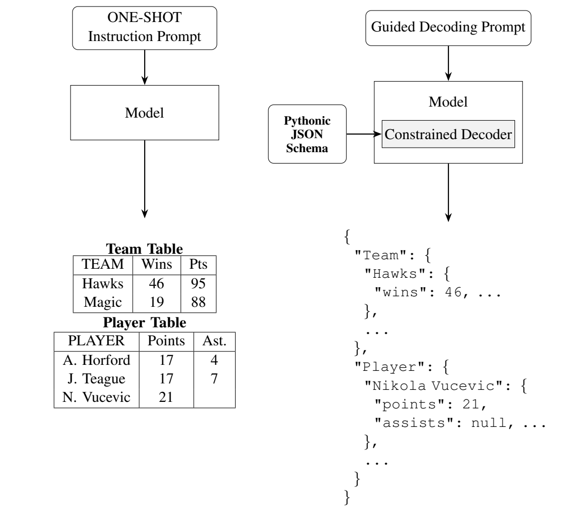
In the free-form approach, LLMs generate a markdown sequence for a given input text. A one-shot instruction prompt, empirically refined through iterative experimentation, encourages the model to adhere to the desired table output format. Specifically, we provide the intended table structure in the prompt, specifying header cells as well

Dataset	Table							Input Text			
	N	Rows min	Rows max	Rows mean	Cols min	Cols max	Cols mean	N	Word min	Word max	Word mean
E2E	4693	2	2	2.00	1	7	5.40	4693	4	71	24.06
RW Teams	687	2	3	2.91	1	9	4.22	728	135	695	308.36
RW Players	724	2	16	7.49	1	17	7.94				
LiveSum	754	3	3	3.00	9	9	9.00	754	724	1760	1138.40

Table 1: Descriptive statistics of the gold table data and the input texts for each dataset. For the Rotowire dataset we show two rows that correspond to the different table types (Teams and Players).

as the number of columns and rows, inspired by Tang et al. (2024).

In contrast, the schema-guided approach enforces tighter structural guarantees through decoding constraints defined by a provided JSON schema. To enable constrained decoding for structured table generation, we implement a schema builder that dynamically constructs a nested JSON schema based on the table layout specified by the row and column headers in the gold data. Cell values are represented as nullable integers, while both row and column headers are constrained to predefined values.



(a) Open Prompt w/ Markdown Tables (b) Guided Decoding w/ Structured JSON

Figure 2: Comparison of open prompting and guided decoding. In guided decoding, the decoder component is constrained to a Pythonic JSON schema.

3.3 Parsing & Post-processing

The postprocessing step of the schema guided approach is comparatively simple. The LLM emits JSON with Pydantic-safe keys, while the schema keeps the true column names in each property's title metadata. We swap every key for its title (e.g., `total_points` → 'Total Points'), then turn each top-level object into a pandas DataFrame.

In case of unstructured (free-form) generation of markdown tables, Large language models (LLMs) often ignore rigid "table-only" instructions. Even when explicitly asked to emit nothing but a Markdown table, they may

- prepend or append free-form prose,
- break a single table with stray blank lines or malformed rows, or
- return *several* distinct tables in succession.

To robustly recover well-formed tabular data under all of these failure modes, we adopt a two-stage, candidate-based pipeline: we first *extract* every pipe-delimited region that *could* be a table, then *validate* each region against Markdown's structural rules. This separation lets us (i) retain all legitimately produced tables, regardless of how many the model generates. This is particularly important for the Rotowire dataset, where the expected output frequently consists of two tables; and (ii) pinpoint exactly where and why a malformed candidate breaks the specification.

1. Candidate extraction. We scan the raw LLM output line-by-line, grouping together each maximal run of lines that begin with a pipe (|). Every such run becomes a *candidate block*: it might be a complete table, a fragment, or just arbitrary pipe-separated text. Because we postpone any judgment of correctness, no genuine table can be missed.

2. Table validation & parsing. Each candidate block is subjected to four sequential regex checks:

(i) **Header integrity.** The first line must start and end with | and contain at least one non-pipe character between consecutive pipes.

(ii) **Separator row.** The second line must also be pipe-delimited and include at least three hyphens per column (optionally flanked by colons), satisfying Markdown's header-body separation rule.

(iii) **Row consistency.** Every subsequent line must open and close with | and, when split on pipes, yield the same number of cells as the header, guaranteeing a uniform column alignment.

(iv) **Table size.** Each table must have at least three rows: One header, one separator and one data row.

Only candidates that pass all four checks are split on pipes, trimmed of whitespace, and assembled into a structured grid (e.g., a `DataFrame`). Failures, such as missing pipes, malformed separators, or mismatched cell counts, raise specific errors, enabling fine-grained diagnostics of LLM formatting bugs.

Because the extraction and validation steps operate on each candidate independently, the pipeline naturally recovers valid tables in their original order, even when a single model invocation produces multiple tables.

3.4 Evaluation

For constructing a mapping between table candidates and gold table we are following a greedy table assignment approach. For each candidate table we are scoring the overlap between its column headers and the gold table using case-insensitive matching. The candidate table with the highest score is then assigned for evaluation and if there is not at least one table with at least one overlapping column header found in the candidates for evaluation, we mark the table as missing.

3.4.1 Metrics

For assessing the quality of the generated tables we report performance at three granularity levels: Cell-Level, Row-Level, and Table-Level. At the strictest level, we measure *table accuracy*, defined as the proportion of tables where every normalized predicted cell exactly matches the corresponding gold cell. At the *row-level*, we count a true positive (TP) as a predicted row that exactly matches a gold row (order-agnostic), a false positive (FP) as an extra predicted row, and a false negative (FN) as a missing gold row—suitable when rows represent unique entities (e.g., player or team statistics). At the *cell-level*, TP occurs when predicted and gold values exactly match after normalization, FP when a predicted cell exists but the gold cell does not, and FN when a gold value exists but the prediction is missing or incorrect. We then report F1 scores at both cell and row levels.

We calculate the ROUGE-L Score (Lin, 2004) and the Levenshtein ratio, a normalized measure derived from Levenshtein distance (Levenshtein, 1966), to quantify string similarity upon table level, after transforming the DataFrame into a table sequence, by deterministically applying a minimal markdown table format. We also calculate the Levenshtein ratio positionally on cell-level and calculate the average over every (non-header) cell, treating missing cells as empty strings.

For the Rotowire and Livesum datasets, we further calculate Root Mean Square Error (RMSE), since their inner cells are numeric. All metrics exclude header cells, as these are always provided by the schema-containing prompts. The metrics are only calculated on tables that were present in the generated output. Missing tables therefore do not influence the calculation per metric, but we keep track of the actual presence of expected tables in the generated outputs.

3.5 Experimental Setup

Our experiments were performed on a single node of the high-performance computing cluster at the scientific computing cluster of the University of Leipzig[1]. The node contains two AMD(R) EPYC(R) 7713 CPUs with 64 cores each, 1TB RAM and eight Nvidia A30 GPUs, each with 24GB HBM2 RAM. For the generation we leverage the `vLLM` library (Kwon et al., 2023), with a unified setup over all evaluated models: A `temperature` of 0.0, a `max_model_len` of 6144 and `max_new_tokens` of 4096. For structured decoding we are using the `xgrammar` package. The code we used for generating and evaluating our models was made available in our Gitlab Repository[2].

4 Results

4.1 E2E

The Cell Level metrics, Levenshtein and F1, show in general higher values than the ones for row and table level. In comparison to Levenshtein and Rouge-L on Table Level we see - apart from the outlier - a clear expressed gain in performance with rising parameter sizes.

[1] https://www.sc.uni-leipzig.de/

[2] https://github.com/JulianOestreich90/text2table

Model	Presence (%)		Cell				Row		Table					
			F1		Levenshtein		F1		Accuracy		Levenshtein		ROUGE-L	
	U	S	U	S	U	S	U	S	U	S	U	S	U	S
Qwen2.5-0.5B-Instruct	99.66	**100.00**	0.724	0.519	0.804	0.753	0.231	0.060	0.217	0.060	0.888	0.864	0.871	0.831
Qwen2.5-1.5B-Instruct	99.34	**100.00**	0.845	0.763	0.912	0.860	0.436	0.295	0.436	0.295	0.943	0.908	0.943	0.898
Qwen2.5-3B-Instruct	99.91	**100.00**	0.822	0.832	0.876	0.906	0.383	0.431	0.381	0.431	0.939	0.930	0.936	0.937
Qwen2.5-7B-Instruct	99.98	**100.00**	0.888	0.787	0.932	0.867	0.561	0.314	0.561	0.314	0.950	0.917	0.954	0.913
Qwen2.5-14B-Instruct	**100.00**	**100.00**	**0.891**	0.850	**0.934**	0.910	**0.570**	0.474	**0.570**	0.474	**0.953**	0.931	**0.957**	0.940
Qwen2.5-32B-Instruct	**100.00**	95.50	0.883	0.728	0.931	0.807	0.531	0.229	0.531	0.229	**0.953**	0.874	0.954	0.861
Falcon3-1B-Instruct	97.49	**100.00**	0.595	0.752	0.672	0.825	0.039	0.197	0.038	0.197	0.877	0.905	0.822	0.888
Falcon3-3B-Instruct	99.49	**100.00**	0.871	0.758	0.920	0.860	0.507	0.279	0.507	0.279	0.949	0.911	0.951	0.903
Falcon3-7B-Instruct	99.96	**100.00**	0.880	0.836	0.932	0.905	0.548	0.443	0.548	0.443	0.952	0.930	0.954	0.933
Falcon3-10B-Instruct	**100.00**	**100.00**	0.881	0.786	0.930	0.866	0.547	0.295	0.546	0.295	0.951	0.917	0.955	0.910
Phi-4-mini-Instruct	**100.00**	**100.00**	0.838	0.707	0.885	0.846	0.439	0.208	0.439	0.208	0.944	0.900	0.944	0.890
Phi-4	**100.00**	**100.00**	0.818	0.727	0.850	0.828	0.401	0.198	0.401	0.198	0.945	0.902	0.946	0.889

Table 2: Evaluation metrics on the E2E test set. For each metric, Unstructured (U) and Structured (S) results are shown with the best model in bold and second best model underlined.

4.2 Rotowire

The evaluation results (Table 3, Table 4) demonstrate that guided (structured) decoding consistently improves model performance on the Rotowire dataset. Across both Team and Player tables, all evaluated models achieve high table presence rates, with structured decoding frequently reaching or approaching 100%. Notably, the smallest model (Qwen2.5-0.5B-Instruct) generates only 43% of player tables in the unstructured setting, but rises to 99.4% with guided decoding. For all core metrics - RMSE, cell F1, cell cell Levenshtein, row F1, table exact match, Levenshtein, and ROUGE-L - structured outputs generally yield equal or higher scores than unstructured ones. The lowest RMSEs are observed for the largest Qwen2.5 models with structured decoding (e.g., 1.78 for Player, 6.05 for Team). Both cell F1 and Lev-F1 scores are maximized in structured outputs of larger Qwen2.5 and Falcon models, frequently exceeding 0.96.

An analysis of the errors that occured, when validating the table candidates in the unstructured outputs show (Figure 3), that by far the most common error type is the column missmatch. For all different model families these error is reduced constantly with an increase of model size, however the Qwen and the Falcon family show a rise of candidate errors for the biggest evaluated models (32B and 10B respectively). When comparing Falcon3-7B and Falcon3-10B, it is to note, that while the table presences only drop by 1%, the amount of candidate errors rises overproportional.

4.3 Livesum

Results on the Livesum dataset (Table 5) indicate that, while structured decoding increases the presence rate always to 100%, it does not consistently improve table quality. For most cell-level metrics, unstructured outputs usually achieve better values. Importantly, none of the generated tables - regardless of decoding strategy - perfectly matches the ground truth, as evidenced by the absence of exact matches at both row and table levels.

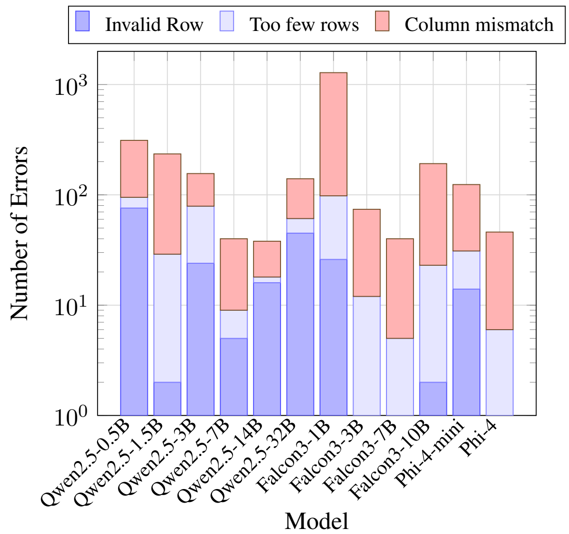


Figure 3: Distribution of errors for parsing the table candidates from unstructured generation on the Rotowire dataset. ‘Invalid row’ errors combine all errors with invalid separator line, invalid headers and invalid data rows. Tables are classified as ‘Too few rows’ with less than 3 rows and ‘Column mismatch’ occurs when the amount of columns over the table rows does not align.

5 Discussion

5.1 Decoding Strategy across Tasks

Our results proved, that structured decoding consistently boosts the *presence* of valid tables across all three benchmarks: malformed outputs are far less

Model	Presence (%)		Cell						Row		Table					
			RMSE		F1		Levenshtein		F1		Exact Match		Lev		ROUGE-L	
	U	S	U	S	U	S	U	S	U	S	U	S	U	S	U	S
Qwen2.5-0.5B-Instruct	98.4	99.7	33.16	35.94	0.597	0.722	0.634	0.719	0.025	0.026	0.010	0.020	0.888	0.937	0.723	0.739
Qwen2.5-1.5B-Instruct	**100.0**	**100.0**	39.57	17.77	0.619	0.859	0.581	0.847	0.042	0.140	0.031	0.090	0.949	0.961	0.874	0.839
Qwen2.5-3B-Instruct	<u>99.9</u>	**100.0**	43.85	14.35	0.595	0.883	0.549	0.853	0.042	0.194	0.031	0.108	0.962	0.964	0.917	0.857
Qwen2.5-7B-Instruct	99.7	**100.0**	38.85	9.72	0.639	0.921	0.563	0.878	0.108	0.346	0.068	0.194	0.972	0.976	0.946	0.904
Qwen2.5-14B-Instruct	**100.0**	**100.0**	33.68	<u>6.33</u>	0.764	<u>0.964</u>	0.705	**0.939**	0.105	<u>0.605</u>	0.038	<u>0.426</u>	0.983	**0.989**	**0.969**	0.959
Qwen2.5-32B-Instruct	96.9	99.4	41.40	**6.05**	0.623	**0.965**	0.540	**0.939**	0.056	**0.616**	0.017	**0.433**	0.964	**0.989**	0.952	0.958
Falcon3-1B-Instruct	96.2	85.7	51.16	37.38	0.370	0.704	0.389	0.780	0.015	0.026	0.015	0.020	0.838	0.923	0.686	0.724
Falcon3-3B-Instruct	98.8	96.5	48.25	30.98	0.501	0.777	0.431	0.824	0.021	0.066	0.019	0.048	0.943	0.946	0.885	0.776
Falcon3-7B-Instruct	**100.0**	**100.0**	35.69	10.39	0.730	0.928	0.670	0.886	0.074	0.394	0.033	0.234	0.980	0.980	0.957	0.923
Falcon3-10B-Instruct	**100.0**	**100.0**	47.07	7.52	0.595	0.951	0.501	<u>0.919</u>	0.031	0.539	0.020	0.374	0.974	<u>0.986</u>	0.966	0.948
Phi-4-mini-Instruct	**100.0**	80.3	33.20	32.53	0.750	0.628	0.693	0.553	0.063	0.082	0.031	0.051	0.975	0.951	0.936	0.807
Phi-4	**100.0**	**100.0**	47.80	6.44	0.675	0.937	0.527	0.899	0.018	0.600	0.015	0.405	0.967	**0.989**	<u>0.968</u>	0.957

Table 3: Evaluation metrics on the Team Tables of the Rotowire test set. For each metric, Unstructured (U) and Structured (S) results are shown with the best model in bold and second best model underlined.

Model	Presence (%)		Cell						Row		Table					
			RMSE		F1		Levenshtein		F1		Accuracy		Levenshtein		ROUGE-L	
	U	S	U	S	U	S	U	S	U	S	U	S	U	S	U	S
Qwen2.5-0.5B-Instruct	43.1	<u>99.4</u>	11.47	10.98	0.661	0.697	0.550	0.633	0.001	0.022	0.000	0.006	0.684	0.925	0.522	0.674
Qwen2.5-1.5B-Instruct	70.9	**100.0**	8.97	7.97	0.736	0.763	0.630	0.683	0.054	0.105	0.004	0.025	0.854	0.936	0.706	0.747
Qwen2.5-3B-Instruct	88.5	**100.0**	6.96	4.97	0.751	0.898	0.655	0.835	0.114	0.270	0.026	0.046	0.909	0.945	0.769	0.798
Qwen2.5-7B-Instruct	95.3	**100.0**	6.19	3.22	0.745	0.938	0.638	0.894	0.133	0.455	0.021	0.112	0.939	0.954	0.846	0.838
Qwen2.5-14B-Instruct	96.3	**100.0**	4.99	<u>2.19</u>	0.837	<u>0.964</u>	0.756	<u>0.934</u>	0.301	<u>0.623</u>	0.061	0.188	0.962	0.955	<u>0.919</u>	0.851
Qwen2.5-32B-Instruct	77.1	<u>99.4</u>	5.12	**1.78**	0.794	**0.971**	0.689	**0.946**	0.223	**0.687**	0.037	**0.224**	0.892	0.957	0.843	0.859
Falcon3-1B-Instruct	35.6	84.1	12.58	14.74	0.649	0.536	0.526	0.532	0.002	0.013	0.001	0.006	0.690	0.872	0.534	0.606
Falcon3-3B-Instruct	92.7	96.7	6.70	7.15	0.663	0.656	0.569	0.585	0.102	0.059	0.015	0.021	0.914	0.933	0.775	0.699
Falcon3-7B-Instruct	95.2	**100.0**	4.79	3.48	0.854	0.924	0.778	0.874	0.271	0.433	0.059	0.119	0.957	0.954	0.879	0.835
Falcon3-10B-Instruct	94.2	**100.0**	5.88	2.41	0.780	0.961	0.673	0.930	0.186	0.609	0.026	0.189	0.940	0.956	0.909	0.851
Phi-4-mini-Instruct	87.2	81.1	4.96	7.30	0.859	0.826	0.779	0.731	0.268	0.203	0.038	0.044	**0.972**	0.952	0.884	0.784
Phi-4	**100.0**	**100.0**	6.11	2.01	0.838	0.956	0.733	0.923	0.164	0.634	0.006	<u>0.196</u>	<u>0.970</u>	0.956	**0.956**	0.852

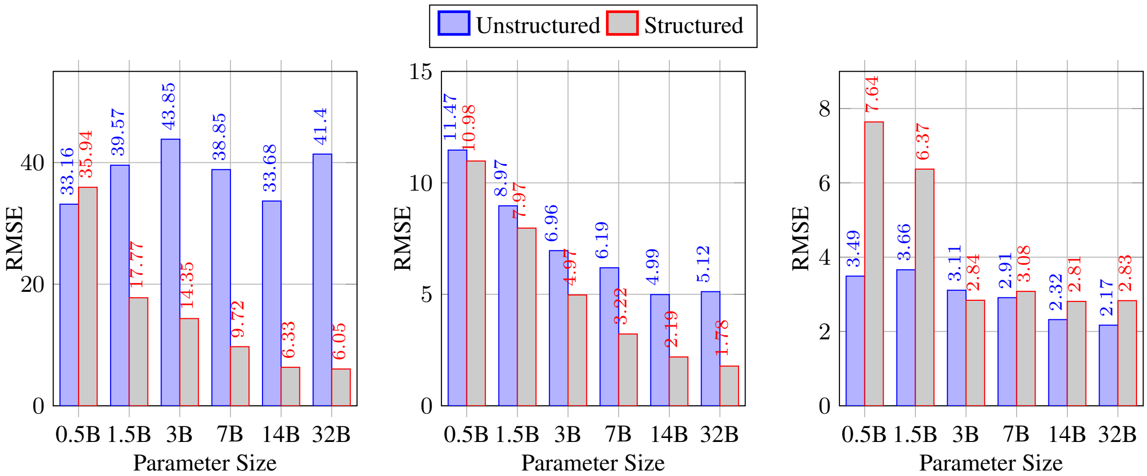
Table 4: Evaluation metrics on the Player tables of the Rotowire dataset for Unstructured (U) vs. Structured (S) generation with the best model in bold and second best model underlined.

common than with one-shot prompting. Beyond mere presence, it improves table *quality* on both Rotowire tasks (Team and Player), whereas it is counterproductive on E2E and Livesum. On the Rotowire benchmark, the schema eliminates most errors: even the smallest model (Qwen2.5-0.5B) shows an increased performance once structured decoding is applied, and the gains grow significantly with parameter size. In contrast, unstructured generation struggles on team tables to reliably align values with the correct entities, likely due to the dense and entangled presentations of statistics for both teams in the source text. The picture flips on the E2E dataset: critical attributes are densely packed within short utterances, and the freedom of unconstrained decoding lets larger models capture subtle lexical cues better than a rigid schema. The results for Livesum show, that even though structured decoding guarantees full coverage of the tables it does also not raise the quality metrics. Here we assume that the reason is due to the high reasoning requiring task and aggregating information over long contexts, a task where already Tam et al. (2024) showed, that reasoning skills are lowered, when enforcing high structural constrained.

In short, strict schema-guided decoding is helpful when numerical information can be directly extracted from sparsely spread information in the text (Rotowire) but can hinder performance when textual information is densely packed (E2E). For Livesum, the models must infer final values by aggregating evidence scattered across long articles, a reasoning-heavy task that neither decoding strategy handles well.

5.2 Influence of Model Size

Overall, we observe a clear positive relationship between model size and the validity and quality of generated tables. However, this trend does not hold uniformly across all datasets and decoding methods. Notably, our largest evaluated model, Qwen2.5-32B, demonstrates superior performance on the Livesum and Rotowire datasets according to most metrics, yet it unexpectedly shows reduced table presence, particularly pronounced on the Rotowire dataset in both decoding settings, and on the E2E dataset with structured decoding. It also showed significantly reduced table quality with re-



(a) Rotowire Team (b) Rotowire Player (c) Livesum

Figure 4: The influence of Qwen parameter size on the RMSE for unstructured vs. structured generation on the different table types with numerical cells.

Model	Presence (%)		Cell						Row		Table					
			RMSE		F1		Levenshtein		F1		Accuracy		Levenshtein		ROUGE-L	
	U	S	U	S	U	S	U	S	U	S	U	S	U	S	U	S
Qwen2.5-0.5B-Instruct	99.7	**100.0**	3.49	7.64	0.481	0.460	0.768	0.755	0.000	0.000	0.000	0.000	0.925	0.922	0.674	0.625
Qwen2.5-1.5B-Instruct	99.6	**100.0**	3.66	6.37	0.549	0.403	0.789	0.516	0.000	0.000	0.000	0.000	0.931	0.936	0.693	0.690
Qwen2.5-3B-Instruct	96.7	**100.0**	3.11	2.84	0.562	0.586	0.798	0.783	0.000	0.000	0.000	0.000	0.932	0.937	0.695	0.714
Qwen2.5-7B-Instruct	98.5	**100.0**	2.91	3.08	0.490	0.571	0.777	0.773	0.000	0.000	0.000	0.000	0.928	0.935	0.683	0.710
Qwen2.5-14B-Instruct	97.1	**100.0**	2.32	2.81	0.647	0.581	0.826	0.727	0.000	0.000	0.000	0.000	0.942	0.938	0.735	0.727
Qwen2.5-32B-Instruct	99.9	**100.0**	**2.17**	2.83	**0.670**	0.573	**0.837**	0.693	**0.002**	0.000	0.000	0.000	**0.944**	0.939	**0.747**	0.728
Falcon3-1B-Instruct	82.1	**100.0**	3.99	7.56	0.440	0.308	0.727	0.713	0.000	0.000	0.000	0.000	0.900	0.879	0.653	0.574
Falcon3-3B-Instruct	92.3	**100.0**	3.42	5.41	0.528	0.557	0.783	0.790	0.000	0.000	0.000	0.000	0.929	0.932	0.690	0.706
Falcon3-7B-Instruct	**100.0**	**100.0**	3.13	5.34	0.577	0.439	0.803	0.515	0.000	0.000	0.000	0.000	0.934	0.924	0.701	0.693
Falcon3-10B-Instruct	**100.0**	**100.0**	2.79	4.27	0.579	0.587	0.803	0.750	0.000	0.000	0.000	0.000	0.934	0.936	0.702	0.728
Phi-4-mini-Instruct	**100.0**	**100.0**	3.47	6.86	0.595	0.472	0.796	0.505	0.000	0.000	0.000	0.000	0.933	0.939	0.718	0.717
Phi-4	**100.0**	**100.0**	2.47	3.01	0.648	0.555	0.826	0.708	0.000	0.000	0.000	0.000	0.942	0.934	0.736	0.720

Table 5: Evaluation metrics on the Livesum test set. For each metric, Unstructured (U) and Structured (S) results are shown with the best model in bold and second best model underlined.

spect to the metrics, than smaller models of the same family on E2E. For tables containing numerical values, we further investigated the RMSE to better reflect true table fidelity, as string-based metrics may provide misleading interpretations here. Larger models typically yielded lower RMSE values, however, the Rotowire Team Tables showed consistently high RMSE under unstructured decoding, regardless of model size, which suggests limitations specific to that scenario. In contrast, structured decoding consistently improved RMSE performance on these tables.

In summary, while larger models generally enhance table generation quality and validity, our work also showed exceptions, particularly Qwen2.5-32B, where increased model size adversely affects table presence and certain performance aspects.

5.3 Suitability of Evaluation Metrics

Our findings indicate a limited suitability of common NLP metrics at the table level. Despite high Levenshtein ratios and ROUGE-L scores, the models never achieve exact row- or table-level matches, indicating that such metrics may overestimate the true quality of the generated tables. A similar observation can be made for finegrained cell level metrics, such as the F1 Score or Cell-level Levenshtein. In contrast, RMSE provides a informative assessment for numerical tables, directly quantifying the deviation from ground truth, but is not applicable to tables with textual cells. The strict exact match metrics at row and table level accurately indicate whether the generated table matches the ground truth, but they fail to account for semantically equivalent variations or minor deviations in string expression. The Levenshtein score on positional cell level has been found useful, as it clearer expresses the differences in performance between the different model sizes, while not being as strict as the exact match metrics. In general the metrics were able to consistently indicate, whether structured decoding performed better or worse on a given benchmark; however, due to their individual limitations, actual table quality is often best assessed through human evaluation.

5.4 Limitations

All our benchmarks assume known schemas for table generation; we do not address open-schema or schema-inference scenarios, as explored in recent work (Ahuja et al., 2025). Our experiments are also limited to a specific one-shot prompt and a structured decoding prompt, and do not consider alternative prompting strategies that might yield better performance. Additionally, we evaluate only a subset of available models, ranging from 0.5B to 32B parameters, from three developers (tiiuae, Qwen, Microsoft), none of whom publish their training data. As a result, we cannot rule out the possibility that some models may have been exposed to our benchmark datasets during training. Furthermore the selected benchmarks represent just a limited range of domains and table types. Our results may not generalize to other datasets, especially those with more complex tables, larger sizes, or more diverse content. Our evaluation also assumes accurate table extraction and preprocessing from ground truth and LLM responses. Any errors or inconsistencies in preprocessing could impact the reported metrics.

6 Conclusion & Future Work

Our study demonstrated that the impact of structured decoding on information extraction in the context of Text-to-Table generations is *highly task dependent*. We derived the following key conclusions:

- **Decoding strategy:** Schema-guided decoding is improving the presence of tables and reducing malformed outputs significantly, but it depresses the table quality where textual facts are densely packed within the input or have to be aggregated over long context.

- **Model size:** The relationship between model size and table generation quality follows predictable trends in most cases, with larger models generally producing higher validity and quality tables. Results on the Rotowire Team tables however show for the unstructured setting, that scale alone does not guarantee optimal performance for table generation tasks.

- **Evaluation Metrics:** String-based NLP metrics on Table level overestimate table quality. While exact match metrics are to strict for probabilistic generated content, they reflect the actual table qualities better. A mixed variant, utilizing NLP based soft-match metrics positionally on cell or row level, seems more promising.

Considering the limitations identified in our work, we recommend several promising directions for future research. First, there is significant value in exploring methods for schema inference, moving beyond the exclusive use of predefined schemas. Further research should also address the generation of more complex tables, such as those exhibiting greater variability in size, multi-line headers, or merged cells. Additionally, we encourage the development of advanced methods for constrained decoding that go beyond the application of standard JSON schemas. For instance, XML-grammars or the design and implementation of table-specific grammars tailored to the target table sequence language, alongside systematic evaluation of their computational efficiency. Finally, we emphasize the need for novel evaluation metrics and Text-to-Table datasets, ideally complemented by human assessments, to more robustly measure the effectiveness of generated tables and better capture the nuances of real-world use cases.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Research, Technology, and Space (BMFTR) under the grant numbers 01IS24037B and 01IS24077A. Computations for this work were done (in part) using resources of the Leipzig University Computing Centre.

References

Naman Ahuja, Fenil Bardoliya, Chitta Baral, and Vivek Gupta. 2025. Mapmake: Schema guided text to table generation. Accepted to ACL 2025.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: describing table region with natural language. 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.

Zhongpu Chen, Yinfeng Liu, Long Shi, Zhi-Jie Wang, Xingyan Chen, Yu Zhao, and Fuji Ren. 2025. Mdeval: Evaluating and enhancing markdown awareness in large language models. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2981–2991, New York, NY, USA. Association for Computing Machinery.

Steven Coyne and Yuxin Dong. 2024. Large language models as generalizable text-to-table systems. In *Proceedings of the 30th Annual Conference of the Association for Natural Language Processing (NLP2024)*, pages 3243–3252. Tohoku University & NEC Corporation.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction. In *EMNLP*, pages 9300–9322.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, Singapore. Association for Computational Linguistics.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore. Association for Computational Linguistics.

Benno Kruit, Hongyu He, and Jacopo Urbani. 2020. Tab2know: Building a knowledge base from tables in scientific papers. In *The Semantic Web – ISWC 2020*, pages 349–365, Cham. Springer International Publishing.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

R. Lebret, D. Grangier, and M. Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain . In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, 76:100761.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D'Antoni. 2025. Grammar-aligned decoding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Mandar Sharma, Ajay Kumar Gogineni, and Naren Ramakrishnan. 2024. Neural methods for data-to-text generation. *ACM Trans. Intell. Syst. Technol.*, 15(5).

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-table: A new way of information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2518–2533, Dublin, Ireland. Association for Computational Linguistics.

Evaluating the LLM and NMT Models in Translating Low-Resourced Languages

Julita Pucinskaite
Lancaster University, United Kingdom
julitapucinskaite@gmail.com

Ruslan Mitkov
University of Alicante, Spain
r.mitkov@lancaster.ac.uk

Abstract

Machine translation has significantly advanced due to the development of transformer architecture, which is utilised by many modern deep-learning models. However, low-resource languages, such as Lithuanian, still face challenges stemming from the limited availability of training data and resource constraints. This study examines the translation capabilities of Neural Machine Translation (NMT) models and Large Language Models (LLMs), comparing their performance in low-resource translation tasks. Furthermore, it assesses the impact of parameter scaling and fine-tuning on their effectiveness in enhancing model performance. The evaluation showed that while LLMs demonstrated proficiency in low-resource translation, their results were lower compared to NMT models, which remained consistent across smaller variants. However, as model size increased, the lead was not as prominent, correlating with automatic and human evaluations. The effort to enhance translation accuracy through fine-tuning proved to be an effective strategy, demonstrating improvements in vocabulary expansion and structural coherence in both architectures. These findings highlight the importance of diverse datasets, comprehensive model design, and fine-tuning techniques in addressing the challenges of low-resourced language translation. This project, one of the first studies to focus on the low-resourced Lithuanian language, aims to contribute to the broader discourse and ongoing efforts to enhance accessibility and inclusivity in Natural Language Processing.

1 Introduction

The field of Natural Language Processing (NLP) has been essential in enhancing access to information and promoting inclusivity across different languages. Machine Translation (MT) was developed to utilise computers in overcoming communication gaps and facilitating cross-linguistic cooperation, with early efforts focusing on translating Russian to English. However, despite significant advancements in LLMs, MT and NLP in general, many low-resourced languages remain underrepresented and overlooked by the rapidly growing AI industry.

It is worth noting that Machine Translation has long been a key focus in NLP with the aim of enabling computers to translate natural language automatically. Initially, the field was dominated by the Rule-Based (RB) approach, which relied on manually constructed linguistic rules and dictionaries. However, this method was prone to error, resource intensive and had scalability implications when transferring rules between different languages (Wang et al., 2022). Due to these limitations, interest in RB systems declined, leading to a slowdown in the progress within the MT field. Nevertheless, some continued, resulting in the development of highly accurate RB systems such as Systran and DeepL, while they later transitioned first to statistical and after that to neural network-based architectures.

The field saw meaningful breakthroughs with the adoption of corpus-based methods following the Statistical Machine Translation (SMT), which was reintroduced in the early 1990s by IBM researchers (Brown et al., 1990). SMT leverages large parallel texts and probabilistic models to make predictions on the most likely translation. Initially, these systems relied on single-word mappings, although this introduced many errors in semantic meaning and word reordering, leading to a shift toward phase-based translation (Lopez,

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 123–133, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-013

2008). This approach was the foundation for an early version of the Google Translate engine. Despite these advancements, SMT struggled with long-distance word ordering and data sparsity issues, particularly for linguistically distant language pairs (Wang et al., 2022).

The introduction of deep learning techniques, such as a sequence-to-sequence model structure, transformed MT. These models, powered by neural networks utilised an encoder-decoder framework that mapped input sentences to variable-length vector representations, ensuring the retention of sentence structure and meaning (Sutskever, 2014). The addition of the attention mechanisms allowed the decoder layer to focus solely on the relevant input encodings, improving translation fluency and overcoming SMT weaknesses (Bahdanau, 2014). This neural process was extended to multilingual machine translation, where shared encoded representations could be supported by multiple decoder layers for different target languages (Dong et al, 2015).

The Transformer model architecture revolutionised NLP by introducing self-attention mechanisms, removing the need to use recurrence and process one token at a time. Unlike earlier models, these properties allow the model to read all tokens simultaneously, capturing broad contextual relationships regardless of sentence length. This parallel processing led to a significantly faster training on large datasets, making Transformers the foundation of NMT models and LLMs (Vaswani, 2017). NMT models follow a sequence-to-sequence framework, mapping an input sequence from the source language to the target language. Where LLMs are typically categorised as auto-encoding or auto-regressive models, either using encoder or decoder-only architectures, with the latter being more frequent and following the objective of accurately predicting the next token in the sequence (Dong et al., 2019).

The performance of LLM and NMT models is dependent on the availability and quality of the training corpus. These models typically rely on high-resourced languages, such as English and German, with low-resourced languages receiving significantly less representation due to data limitations (Scao et al., 2022). NMT models are usually pre-trained on parallel corpora, which enables a comprehensive representation of language distribution. In contrast, LLMs are trained on diverse texts without targeting multilingualism, which often limits their ability to support low-resource tasks (Paupard, 2024).

To address this disparity, researchers reinforced insufficient parallel corpora with monolingual data (Znang and Zong, 2016). However, a high monolingual data ratio can diminish models learning outcomes, calling for back-translation, which automatically incorporates translations to monolingual texts (Sennrich et al., 2015).

An equally critical aspect is dataset quality, particularly in low-resource settings. A study found that noisy texts can drastically degrade translation accuracy, making cleaned and filtered datasets essential for reliable training (Khayrallah and Koehn et al., 2018). This is highly relevant for underrepresented languages, where datasets are often accumulated using web scraping techniques such as Common Crawl, which collect texts from various internet sources (Toral et al., 2017; Baack, 2024). These findings highlight the key challenges for both NMT models and LLMs that require high volumes of training data but are constrained to limited, low-quality, low-resourced language texts.

The open source and distillation techniques seek to bridge this gap and support a transparent and community-driven development process to direct a more inclusive and comprehensive language technology (White et al., 2024). While advocates for the closed-source design argue that it offers better security and data protection guarantees (Xi, 2025). Despite these claims, closed-source models remain vulnerable to various security risks, including adversarial attacks, suggesting that their motivations may be ineffective (Das et al., 2025).

To utilise both open-source accessibility and closed-source performance, researchers have turned to knowledge distillation, where smaller student models learn from larger teacher models. This technique reduces computational demands while ensuring high accuracy and maintainability of core capabilities (Hsieh et al., 2023). The effectiveness was demonstrated by models like Deepseek, which outperformed state-of-the-art models in multiple evaluation benchmarks (Deepseek-AI, 2025).

The present study will experiment with the distilled versions of both NMT and LLM, including NLLB and Gemma models (Costa-Jussà et al., 2022; Team et al., 2024). Although NLLB is fully open-sourced, Gemma follows an open-weights approach where only the model's parameters are made available without source code. While not as transparent as open-source, this still enables customisation and adaptation, supporting resource-constrained teams working on low-resourced language tasks (Zhao et al., 2023).

Finally, it is worth noting that the field of Low-Resource NLP has gained significant attention in recent years, as demonstrated by the growing number of research contributions addressing data scarcity and model adaptability challenges, further emphasising the need to improve machine translation for languages like Lithuanian (Pakray, 2025). The present study is significant in highlighting the insufficient support and inclusion of Lithuanian, a low-resourced language, in modern deep-learning tools and LLMs, an area of study that has received little attention. Researchers in developing translation models often neglect the underrepresented languages due to the limited availability of parallel corpora, which are essential for training accurate translation systems (Chakravarthi et al., 2019). As a result, models trained on small or insufficiently diverse datasets often produce inaccurate translations and hallucinations (Poupard, 2024).

The findings from this study aim to contribute to the enhancement of translation technology, making NLP tools more inclusive and accessible for speakers of less commonly spoken languages. Furthermore, by understanding the limitations of pre-trained models and the benefits of fine-tuning, this research can provide insights in directing future machine translation efforts for other low-resource languages.

The rest of the paper is structured as follows: Section 2 outlines related work, Section 3 presents the methodology and Section 4 provides evaluation results, discussion and error analysis. Finally, Section 5 summarises the work with a conclusion.

2 Related Work

Several studies have addressed the disparity in lesser-spoken languages by developing more linguistically inclusive models. The No Language Left Behind (NLLB) team supported over 200 underrepresented languages by training a multilingual NMT model on high-quality parallel and monolingual datasets and adopting self-supervised learning. These unconventional training methods demonstrated enhanced translation performance in low-resource settings even for languages where explicit training was not undertaken (Costa-Jussà et al., 2022).

Martins et al. (2024) trained the EuroLLM model to address the lack of open-weight LLMs for European languages. The authors used a parallel corpus that included nearly an equal number of English and non-English representations. Their findings indicated that carefully curated datasets and a custom tokeniser enabled the model to outperform much larger competitors in translation tasks.

Nakvosas et al. (2024) discussed the insufficient number of Lithuanian language tokens in the Llama model. They employed a supervised fine-tuning (SFT) technique to improve the model's performance in English-Lithuanian tasks. This approach involved training a pre-existing model on a high-quality custom dataset, allowing it to enhance its learning and generation accuracy, especially in handling previously unseen data (Church et al., 2021).

Another fundamental challenge is the high computational costs associated with training LLMs and NMT models as they often contain billions of parameters, requiring extensive memory, storage and processing power (Hadi et al., 2023). These demands create significant barriers for smaller research teams, especially in underrepresented linguistic communities.

To address resource constraints, researchers have explored performance-efficient fine-tuning (PEFT) techniques. One widely adopted approach is quantisation, which reduces the precision of model parameters (e.g., to 8-bit or 4-bit), lowering memory usage without experiencing major performance loss (Dettmers et al., 2024). Low-Rank Adaptation (LoRA) further optimises resource requirements by applying fine-tuning only to targeted layers, preserving strong multilingual performance while reducing trainable parameters (Hu et al., 2021). These techniques provide effective solutions to optimise resource usage, democratising access to LLMs and NMT models for low-resourced language researchers.

Finally, a key research question is whether LLMs can match or surpass well-established NMT models in low-resource language translation. While multilingual LLMs such as Gemma and Llama have demonstrated effectiveness in high-resourced translation tasks, achievements in low-resourced languages, such as Lithuanian, have often remained undiscovered. Furthermore, LLM architecture may suffer from accuracy loss and hallucinations, where models generate fabricated information when handling large multilingual datasets (Dong, 2024).

Further research is essential to assess the genuine performance of LLMs on underrepresented languages and to determine the trade-offs between model size and fine-tuning in translation quality, thereby contributing to more inclusive NLP systems.

3 Methodology

The adopted methodology, detailed in this section seeks to reply to the following questions:

1. How do pre-trained LLMs and NMT models perform in low-resourced language translation?
2. Does fine-tuning improve translation accuracy? Is it comparable to parameter scaling?

In particular, we outline the data, models and evaluation methods employed in this study and acknowledge experimental limitations.

3.1 Research Design and Data Collection

This study follows an empirical, quantitative approach to evaluate model performance. Models, datasets and fine-tuning tools were obtained through the Transformers library, which provides open-access NLP resources.

Supervised Fine-tuning (SFT) requires rich translation examples. Although scaling laws suggest that the optimal dataset size should be proportional to the model's parameter number. For example, a 1.3 billion parameter model (NLLB) would need around 30 million sentences (Hoffman et al., 2022). However, further recent research shows that smaller, diverse datasets can still yield sufficient performance (Oliver and Wang, 2024; Zoph et al., 2022).

Given resource limitations and limited text availability, a dataset of 300,000 English-Lithuanian sentence pairs was compiled from the following corpora:

Medical Corpus – domain-specific translations with complex terminology.

Parliamentary Corpus – structured sentence pairs from official proceedings.

Common Crawl Corpus – public web data, cleaned to remove foreign tokens, short or ungrammatical sentences.

Wikipedia Corpus – verified translated sentences from Wikipedia resources.

3.2 Model Choice

The selection of models was guided by open-source or open-weights availability to avoid licensing constraints and facilitate further development. Additionally, given memory constraints, around 2 billion parameter models were selected.

Gemma – Google's lightweight Gemini-based LLM for broad NLP tasks (Team et al., 2024).

EuroLLM – Unbabel's LLM, optimised for multilingual translation tasks across European languages (Martins et al., 2024).

Salamandra – BSC-LT's LLM, focused on European languages (Gonzalez-Agirre et al, 2025).

NLLB – Meta's NMT model, covering 200 low-resourced languages (Costa-Jussà et al., 2022).

Helsinki – NMT model specialised in Baltic languages, ideal for Lithuanian translation (Tiedemann et al., 2024).

Madlad – Google's NMT model supporting 400 languages (Kudugunta et al., 2023).

3.3 Evaluation Metrics

Model performance was quantitatively evaluated at two stages: baseline (pre-trained) and post-SFT. The following automatic metrics were used:

SacreBLEU – an improved version of BLUE, measuring n-grams overlap but limited in semantic meaning and synonyms (Papineni et al., 2002).

CHRF – based on character-level n-gram overlaps, effective for morphologically rich languages and correlating with human judgement (Popović, 2015; Lee et al., 2023).

ROUGE – evaluates precision and quality by measuring unigrams, bigrams, and sequence overlap (Lin and Och, 2004).

METEOR – enhances BLEU by considering synonym matching, stemming, and recall, accounting for a better semantic alignment (Banerjee and Lavie, 2005).

3.4 Human Evaluation

Translations were manually assessed on accuracy, fluency, and appropriateness, following Freitag et al. (2021) guidelines and scored from 1 (very poor) to 5 (excellent). Due to the time constraints, only a subset of sentences was evaluated that covered scientific, official, and casual contexts, with an emphasis placed on semantic ambiguity and metaphorical language. The aim of human evaluation was to identify the strengths and weaknesses of each model in producing grammatically correct and contextually relevant translations.

3.5 Model Evaluation

Each model was configured to correctly handle source and target languages. NMT models require explicit language identifiers, such as appending a prefix to the input sentence for Helsinki. While LLMs are more general-purpose and use a prompt-based format. For EuroLLM, source and target prefixes were needed, where Gemma used special tokens for the start and end of inputs and responses.

Models translated 100 unique test sequences from the Flores+ dataset. The Transformers library was used for tokenisation and inference. Generated translations were decoded and compared using BLEU, METEOR, CHRF and ROUGE.

To assess the impact of model size, both ~2B and larger models (up to 9B parameters) were compared, excluding NLLB and Helsinki, as larger versions were not available. Apart from applying quantisation (4-bit) for efficiency, the evaluation process remained consistent with the previous step. Aiming to determine whether increasing model size shows improvement in translation quality.

3.6 Statistical and Practical Significance

To verify whether the differences in model performances were meaningful, t-scores were calculated for each metric using the formula: $t - score = \frac{value-mean}{standard\ deviation}.$

Given a small sample size (6) and targeting a 95% confidence level, a t-critical value of 2.571 was used. Scores exceeding this threshold were considered to have a statistically significant difference (Benjamin et al., 2018).

Furthermore, to complement statistical significance, Cohen's d effect was used to evaluate the practical significance based on the formula: $Cohen's\ d = \frac{mean1-mean2}{standard\ deviation}$

Providing the magnitude of the differences. Effects of up to 0.5 are considered small to medium, while values >0.8 indicate a strong effect (Gignac and Szodorai, 2016). Together, these measures ensure a robust and comprehensive interpretation of model performance differences.

3.7 Fine-Tuning Models

Due to high resource demands, fine-tuning focused on ~2B parameter models, utilising memory-efficient techniques. Models were quantised to 4-bit and fine-tuned with LoRA, targeting attention and feed-forward layers to reduce overhead while preserving performance.

Training used small batch sizes, combined with gradient accumulation, 2e-4 learning rate with linear scheduling, and Adam optimiser to produce gradual and efficient convergence. Models were evaluated consistently every 500 or 1,000 training steps with 100 test-set sentences separated from prior training and utilising the same metrics as in the baseline evaluation phase. This iterative process ensured steady performance monitoring and allowed parameter adjustments as needed.

4 Evaluation Results, Discussion and Error Analysis

4.1 Performance Comparison with Automatic Metrics

The pre-trained NMT models (Madlad, NLLB, Helsinki) generally outperformed LLMs (EuroLLM, Salamandra, Gemma).

Madlad presented the best BLEU, CHRF and overall scores, indicating strong alignment with reference translations. NLLB followed closely, maintaining a good balance between lexical accuracy and semantic variation (high METEOR). Helsinki performed well at the character-level (CHRF) despite a lower BLEU score. EuroLLM led amongst LLMs with relatively higher BLEU and METEOR scores. Gemma achieved the lowest overall scores with poor BLEU and ROUGE results, suggesting minimal overlap and improper sentence structure.

These findings point to NMT models being better suited for machine translation than LLMs.

Models	BLEU	METEOR	CHRF	ROUGE
Madlad	28	0.55	60	0.55
NLLB	26	0.52	58	0.52
Helsinki	21	0.48	55	0.48
EuroLM	19	0.42	49	0.43
Salaman.	17	0.41	50	0.42

Table 1: Pre-trained model evaluation results.

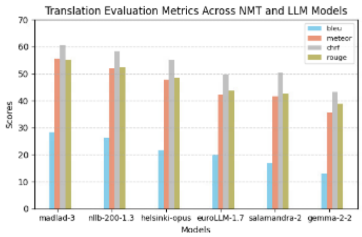


Figure 1: Pre-trained model evaluation.

4.2 Statistical and Practical Significance

T-scores showed that while Madlad consistently performed best and Gemma worst, neither deviated significantly from the mean, not exceeding the 2.571 threshold of 95% confidence. This indicates no statistical performance difference among models.

However, Cohen's d revealed strong practical differences contradicting the t-score. The effect sizes between the best- and worst-performing models, Madlad and Gemma, were on average 2.60 across all metrics, well above the 0.8 threshold for a large effect. Despite not reaching statistical significance, the practical performance difference was considerable.

4.3 Model Size Comparison

Larger models (~9B parameters) demonstrated consistent performance gain over their smaller variants (~2B), raising BLUE scores by 3-5 points while METEOR gains showed more variability, ranging from 0.03 to 0.12 points. The performance increase was more noticeable in LLMs, where NMT models benefited less from scaling up.

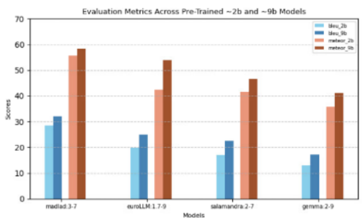


Figure 2: Pre-trained and Scaled Comparison.

Model	bleu_2	bleu_9	meteor_2	meteor_9
Madlad	28	31	0.55	0.58
EuroLLM	19	24	0.42	0.54
Salaman	17	22	0.41	0.46
Gemma	13	17	0.35	0.41

Table 2: Measures of Pre-trained and Scaled Models.

4.4 Pre-trained and Fine-tuned Comparison

Supervised fine-tuning demonstrated clear improvements across all models. BLEU scores rose by 5-8 points, with Madlad gaining 5 and Gemma 8. METEOR improved by 0.04-0.13, with the largest gains observed in LLMs (EuroLLM +0.10, Gemma +0.13).

While NMT models led with strong baseline performance, they showed moderate improvement. In contrast, LLMs started with lower scores but presented comparably larger gains, narrowing the performance gap. Overall, fine-tuning had the strongest impact on LLMs, significantly enhancing their translation quality.

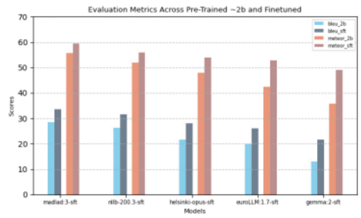


Figure 3: Pre-trained and Fine-tuned Comparison

Models	BLEU	METEOR	CHRF	ROUGE
Madlad	28	0.55	60	0.55
Madlad-lt	33	0.59	62	0.58
NLLB	26	0.52	58	0.52
NLLB-lt	31	0.55	60	0.55
Helsinki	21	0.48	55	0.49
Helsinki-lt	28	0.53	58	0.55
EuroLM	19	0.42	49	0.43
EuroLM-lt	26	0.52	62	0.55
Gemma	13	0.35	43	0.39
Gemma-lt	21	0.48	59	0.53

Table 3: Measures of Pre-trained and Fine-tuned.

4.5 Comparison of Fine-tuning and Scaling

Fine-tuning small models (~2B) led to substantial gains (BLEU +5-8, METEOR +0.04-0.13). However, compared to larger pre-trained models (~9B), fine-tuned models saw smaller performance differences (BLEU +1-4, METEOR +0.02-0.07).

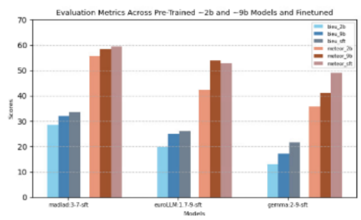


Figure 4: Scaled and Fine-tuning Comparison.

Model	bleu_9	bleu_ft	meteor_9	meteor_ft
Madlad	31	33	0.58	0.59
EuroLLM	24	26	0.54	0.52
Gemma	17	21	0.41	0.48

Table 4: Measures of Scaling and Fine-tuning.

4.6 Discussion on Pre-trained Model Evaluation

Automatic evaluation demonstrated that NMT models consistently outperformed LLMs in translation tasks, highlighting their domain-specific optimisation and better semantic handling. Madlad and NLLB taking the lead across all models, which could be attributed to their extensive multilingual capabilities, enabling broader linguistic variation and generalisation through parameter sharing (Pires et al., 2019). While the underlying success factor is the emphasis on data quality, where Madlad's team prioritised manual auditing, while NLLB curated custom corpus (Kudugunta et al., 2023; Costa-Jussà et al., 2022). Remarkably, NLLB achieved nearly identical results to Madlad despite having half the parameters, likely due to the use of back-translation and knowledge distillation techniques. Helsinki, despite being smaller (<1B), surpassed all LLMs, benefiting from its specialisation in Baltic languages. Nevertheless, it still trailed behind Madlad and NLLB, possibly because it was developed with limited resources compared to other company-backed models. EuroLLM and Salamandra performed comparably to Helsinki, showing that smaller LLMs can achieve competitive performance when designed with a task-specific focus and an emphasis on a high-quality, diverse dataset. Finally, Gemma produced the weakest results, despite its equivalent size and utilisation of distillation, likely caused by its English-focused training, which lacked multilingual depth (Team et al., 2024).

4.7 Discussion on Statistical and Practical Significance

The lack of statistical significance in the t-score could be attributed to the small sample size, which increased the probability of type II error (Huang, 2017). In comparison, Cohen's d practical value revealed a large effect (2.5) between best and worst models and a small effect between close performers (e.g., EuroLLM – Salamandra at 0.12), aligning with automatic evaluations. Though threshold values are estimated and may not be universal (Corell et al., 2020), when used alongside other evaluation methods, they reinforce the reliability of the study's findings.

4.8 Error Analysis with Human Evaluation

Models varied in accuracy with most common mistakes including literal translations of idioms e.g. "field" was translated as "physical location" instead of "area of research or "shine a light" lost its metaphorical meaning. Mistranslations of uncommon terms such as "rabid dog" was interpreted by Gemma as "red dog", while Helsinki presented “rabid” as “rabin”. Hallucinations were observed from LLMs, particularly from EuroLLM, which regularly appended incorrect dates.

Fluency issues were widespread, with grammar being the most common error, with models using incorrect suffixes, verb tenses and pronouns. Notably, Gemma occasionally repeated words or used basic synonyms, showing limited vocabulary.

Appropriateness, which considers contextual and cultural relevance, proved that most models lacked official, scientific or field-specific terminologies and often reused English phrases.

Madlad: Strong domain-specific terms, though making minor grammatical errors. NLLB: Promising results but prone to ungrammatical and inaccurate terminology. Helsinki: Performed poorly with often mistranslations and Anglicisms. EuroLLM: Preserved the intent but suffered from hallucinations (e.g. added dates). Salamandra: Better than Gemma but had a limited vocabulary and common mistranslations. Gemma: Weakest among all models with frequent grammatical and terminology errors or untranslated phrases.

Models	Accur.	Flue.	Appr.	Total
Madlad	4	4	5	**4**
NLLB	3	4	4	**4**
EuroLLM	3	4	4	4
Helsinki	3	3	3	**3**
Salaman.	3	3	3	**3**
Gemma	2	2	2	**2**

Figure 5: Measures of Human Evaluation.

4.9 Discussion on Error Analysis

Human evaluation largely reinforced the automatic metrics rankings, while identifying overlooked word-level mismatches and error patterns. Where Madlad and NLLB correlated to automatic evaluation, while Helsinki and EuroLLM diverged. EuroLLM exhibited frequent hallucinations, whereas Helsinki was more accurate but struggled with domain-specific terms. Gemma performed the worst with weak grammatical comprehension and word repetitions. These insights highlight the need for improved grammatical accuracy, contextual awareness, and vocabulary breadth in Lithuanian.

4.10 Overall Discussion of Results

Performance collectively improved with model size, while NMT models, due to specialised design and insufficient datasets, face a plateau in scaling effect (Kaplan et al., 2020; Ghorbani et al., 2021). In contrast, LLMs showed distinct improvement, accentuating better generalisation with increased parameters (Wei et al., 2024). However, scaling is a resource-intensive choice, making it impractical for low-funded research (Whittaker, 2021).

Fine-tuning offers a cost-effective alternative, significantly increasing smaller models' performance, especially LLMs, by enhancing the output's structure and coherence. Regardless of these benefits, this process has undesirable drawbacks, like concept forgetting, dependency on data quality and overfitting after a certain point (Mukhoti et al., 2023; Dodge et al., 2020). These issues are especially concerning in low-resource languages with limited diversity and quality data.

Moreover, smaller models (<2B) often lack sufficient multilingual representations (Conneau et al., 2020). While parameter-efficient methods such as LoRA can help, they cannot fully compensate for the advantages offered by large-scale models (Pfeiffer et al., 2020). Therefore, fine-tuning improves performance but does not overcome the inherent limitations of smaller models.

5 Conclusion

5.1 Research Limitations

This study was constrained by 12GB VRAM GPU (UcrelHex, 2024), which restricted the ability to fine-tune or evaluate larger models. Access to more powerful hardware may have yielded different results. Additionally, the focus on open-source/open-weight models ensured transparency and accessibility but excluded closed-source alternatives, possibly limiting performance range.

5.2 Future Work

Future research should prioritise expanding resources for low-resource language communities, as emphasised by the NLLB project, which focused on dataset collection before model design. With the use of distillation, to ensure efficiency, however, this process is limited by its knowledge retention and alternatives such as the Mixture of Experts (MoE) framework show promise by activating only the relevant networks, supporting scalability without increasing computational costs (Koishekenov et al., 2022).

Furthermore, advocating for open-source models is essential in supporting ethical, inclusive and transparent NLP research, especially in underrepresented languages. However, many high-performing models remain closed-source, limiting accessibility and collaboration (Worth et al., 2024).

As model architectures evolve, a clearer classification standard is needed as inconsistencies between model labelling complicate comparisons. Less ambiguous categorisation would enhance transparency and rationalise future research.

5.3 Overall Conclusion

This research evaluated LLMs and NMT models' performance in translating into Lithuanian, a low-resourced language, and revealed consistent outperformance of small NMT models compared to similarly sized LLMs. However, after scaling models (~7-9B parameters), higher performance gains were observed with LLMs, suggesting their better generalisation abilities while NMT models remain more efficient for translation tasks within resourced-constrained settings. Additionally, fine-tuning significantly enhances translation quality, introducing trade-offs as potential knowledge loss.

Ultimately, the key barriers to expanding translation capabilities for underrepresented languages remain computational constraints and data availability. Addressing these challenges requires continued investment in multilingual datasets and efficient training methods for building inclusive and reliable translation systems.

References

Baack, S., 2024, June. A critical analysis of the largest source for generative ai training data: Common crawl. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (pp. 2199-2208). https://dl.acm.org/doi/10.1145/3630106.3659033

Bahdanau, D., 2014. Neural machine translation by jointly learning to align and translate. https://arxiv.org/abs/1409.0473

Banerjee, S. and Lavie, A., 2005, June. 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 (pp. 65-72). https://dl.acm.org/doi/10.5555/1626355.1626389

Benjamin, D.J., Berger, J.O., Johannesson, M., Nosek, B.A., Wagenmakers, E.J., Berk, R., Bollen, K.A., Brembs, B., Brown, L., Camerer, C. and Cesarini, D., 2018. Redefine statistical significance. Nature human behaviour, 2(1), pp.6-10. https://doi:10.1038/s41562-017-0189-z

Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J., Mercer, R.L. and Roossin, P.S., 1990. A statistical approach to machine translation. Computational linguistics, 16(2), pp.79-85. https://aclanthology.org/J90-2002/

Chakravarthi, R., B., Arcan, M., and McCrae, J., P., 2019 Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages. In 2nd Conference on Language, Data and Knowledge (LDK 2019). Open Access Series in Informatics (OASIcs), Volume 70, pp. 6:1-6:14. https://doi.org/10.4230/OASIcs.LDK.2019.6

Church, K.W., Chen, Z. and Ma, Y. (2021) 'Emerging trends: A gentle introduction to fine-tuning', Natural Language Engineering, 27(6), pp. 763-778. https://doi:10.1017/S1351324921000322

Correll, J., Mellinger, C., McClelland, G.H. and Judd, C.M., 2020. Avoid Cohen's 'small','medium', and 'large'for power analysis. Trends in cognitive sciences, 24(3), pp.200-207. https://doi.org/10.1016/j.tics.2019.12.009

Costa-jussà, M. R., et al. (2022). No Language Left Behind: Scaling Machine Translation for Low-Resource Languages. Proceedings of the 2022 Annual Conference on Neural Information Processing Systems. https://arxiv.org/abs/2207.04672

Das, B.C., Amini, M.H. and Wu, Y., 2025. Security and privacy challenges of large language models: A survey. ACM Computing Surveys, 57(6), pp.1-39. https://arxiv.org/abs/2402.00888

Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L., 2024. Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems, 36. https://arxiv.org/abs/2305.14314

Dong, D., Wu, H., He, W., Yu, D. and Wang, H., 2015, July. Multi-task learning for multiple language translation. https://aclanthology.org/P15-1166/

Dong, J., 2024. Natural Language Processing Pretraining Language Model for Computer Intelligent Recognition Technology. ACM Trans. Asian Low-Resour. Lang. Inf. Process. https://doi.org/10.1145/3605210

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., 2019. Unified language model pre-training for natural language understanding and generation. Advances in neural information processing systems, 32. https://arxiv.org/abs/1905.03197

Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., Chelba, C. and Cherry, C., 2021. Scaling laws for neural machine translation. https://arxiv.org/abs/2109.07740

Gignac, G.E. and Szodorai, E.T., 2016. Effect size guidelines for individual differences researchers. Personality and individual differences, 102, pp.74-78. https://doi.org/10.1016/j.paid.2016.06.069

Gonzalez-Agirre, A., Pàmies, M., Llop, J., Baucells, I., Da Dalt, S., Tamayo, D., Saiz, J.J., Espuña, F., Prats, J., Aula-Blasco, J. and Mina, M., 2025. Salamandra Technical Report. https://arxiv.org/html/2502.08489v2

Hsieh, C.Y., Li, C.L., Yeh, C.K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.Y. and Pfister, T., 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. https://arxiv.org/abs/2305.02301

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W., 2021. Lora: Low-rank adaptation of large language models. https://arxiv.org/abs/2106.09685

Huang, H., 2017. Uncertainty estimation with a small number of measurements, part I: new insights on the t-interval method and its limitations. Measurement Science and Technology, 29(1), p.015004. DOI:10.1088/1361-6501/aa96c7

Koehn, P., Khayrallah, H., Heafield, K. and Forcada, M.L., 2018, October. Findings of the WMT 2018 shared task on parallel corpus filtering. In EMNLP 2018 Third Conference on Machine Translation (WMT18) (pp. 726-739). https://aclanthology.org/W18-6453/

Koishekenov, Y., Berard, A. and Nikoulina, V., 2022. Memory-efficient nllb-200: Language-specific expert pruning of a massively multilingual machine translation model. *https://aclanthology.org/2023.acl-long.198/*

Kudugunta, S., Caswell, I., Zhang, B., Garcia, X., Xin, D., Kusupati, A., Stella, R., Bapna, A. and Firat, O., 2023. Madlad-400: A multilingual and document-level large audited dataset. Advances in Neural Information Processing Systems, 36, pp.67284-67296. https://arxiv.org/abs/2309.04662

Kudugunta, S., Huang, Y., Bapna, A., Krikun, M., Lepikhin, D., Luong, M.T. and Firat, O., 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. *https://arxiv.org/abs/2110.03742*

Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., Koo, S. and Lim, H., 2023. A survey on evaluation metrics for machine translation. Mathematics, 11(4), p.1006. https://doi.org/10.3390/math11041006

Lin, C.Y. and Och, F.J., 2004, June. Looking for a few good metrics: ROUGE and its evaluation. In Ntcir workshop. https://aclanthology.org/W04-1013.pdf

Lopez, A., 2008. Statistical machine translation. ACM Computing Surveys (CSUR), 40(3), pp.1-49. https://doi.org/10.1145/1380584.1380586

Martins, P.H., Fernandes, P., Alves, J., Guerreiro, N.M., Rei, R., Alves, D.M., Pombal, J., Farajian, A., Faysse, M., Klimaszewski, M. and Colombo, P., 2025. Eurollm: Multilingual language models for europe. Procedia Computer Science, 255, pp.53-62. https://arxiv.org/abs/2409.16235

Mukhoti, J., Gal, Y., Torr, P.H. and Dokania, P.K., 2023. Fine-tuning can cripple your foundation model; preserving features may be the solution. *https://arxiv.org/abs/2308.13320*

Nakvosas, A., Daniušis, P., Mulevicius, V., 2024. Open Llama2 Model For The Lithuanian Language. A Preprint. Neurotechnology. https://arxiv.org/html/2408.12963v1

NLLB Team (2022) No Language Left Behind: Scaling Human-Centered Machine Translation. arXiv preprint arXiv:2207.04672. Available at: https://arxiv.org/abs/2207.04672

Pakray, P., Gelbukh, A. and Bandyopadhyay, S. (2025) 'Preface: Special issue on Natural Language Processing applications for low-resource languages', Natural Language Processing, 31(2), pp. 181–182. doi:10.1017/nlp.2024.34.

Papineni, K., Roukos, S., Ward, T., and Zhu, W., Z., 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02). Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135

Pires, T., Schlinger, E. and Garrette, D., 2019. How multilingual is multilingual BERT?. https://aclanthology.org/P19-1493/

Popović, M., 2015, September. chrF: character n-gram F-score for automatic MT evaluation. https://aclanthology.org/W15-3049/

Poupard, D., 2024. Attention is all low-resource languages need. Translation Studies, 17(2), pp. 424–427. https://doi.org/10.1080/14781700.2024.2336000

Sennrich, R., Haddow, B. and Birch, A., 2015. Improving neural machine translation models with monolingual data. https://aclanthology.org/P16-1009/

Sutskever, I., 2014. Sequence to Sequence Learning with Neural Networks. https://arxiv.org/abs/1409.3215

Team, G., Riviere, M., Pathak, S., Sessa, P.G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A. and Ferret, J., 2024. Gemma 2: Improving open language models at a practical size. https://arxiv.org/abs/2408.00118

Tiedemann, J., Aulamo, M., Bakshandaeva, D., Boggia, M., Grönroos, S.A., Nieminen, T., Raganato, A., Scherrer, Y., Vázquez, R. and Virpioja, S., 2024. Democratizing neural machine translation with OPUS-MT. https://arxiv.org/abs/2212.01936

Toral, A., Esplá-Gomis, M., Klubička, F. et al. Crawl and crowd to bring machine translation to under-resourced languages. Lang Resources & Evaluation 51, 1019–1051 (2017). https://doi.org/10.1007/s10579-016-9363-6

Vaswani, A., 2017. Attention is all you need. Advances in Neural Information Processing Systems. https://arxiv.org/abs/1706.03762

Vidler, J. and Rayson, P. (2024) *UCREL - Hex: A shared, hybrid multiprocessor system*. https://github.com/UCREL/hex

Wang, H., Wu, H., He, Z., Huang, L. and Church, K.W., 2022. Progress in machine translation. Engineering, 18, pp.143-153.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, QV., Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems. https://arxiv.org/abs/2201.11903

White, M., Haddad, I., Osborne, C., Yanglet, X.Y.L., Abdelmonsef, A. and Varghese, S., 2024. The model openness framework: Promoting completeness and openness for reproducibility, transparency, and usability in artificial intelligence. https://arxiv.org/abs/2403.13784

Whittaker, M., 2021. The steep cost of capture. Interactions, 28(6), pp.50-55.

Worth, S., Snaith, B., Das, A., Thuermer, G. and Simperl, E., 2024. AI data transparency: an exploration through the lens of AI incidents. *https://arxiv.org/abs/2409.03307*

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. and Du, Y., 2023. A survey of large language models. https://arxiv.org/abs/2303.18223

KGEIR: Knowledge Graph-Enhanced Iterative Reasoning for Multi-Hop Question Answering

Tianda Sun, Dimitar Kazakov
University of York
{tianda.sun, dimitar.kazakov}@york.ac.uk

Abstract

Multi-hop question answering (MHQA) requires systems to retrieve and connect information across multiple documents, a task where large language models often struggle. We introduce Knowledge Graph-Enhanced Iterative Reasoning (KGEIR), a framework that dynamically constructs and refines knowledge graphs during question answering to enhance multi-hop reasoning. KGEIR identifies key entities from questions, builds an initial graph from retrieved paragraphs, reasons over this structure, identifies information gaps, and iteratively retrieves additional context to refine the graph until sufficient information is gathered. Evaluations on HotpotQA, 2WikiMultiHopQA, and MuSiQue benchmarks show competitive or superior performance to state-of-the-art methods. Ablation studies confirm that structured knowledge representations significantly outperform traditional prompting approaches like Chain-of-Thought and Tree-of-Thought. KGEIR's ability to explicitly model entity relationships while addressing information gaps through targeted retrieval offers a promising direction for integrating symbolic and neural approaches to complex reasoning tasks. Details of the project and the code are published at https://github.com/TiandaSun/KGEIR

1 Introduction

Multi-hop question answering (MHQA) presents a significant challenge in natural language processing, requiring systems to retrieve and connect information from multiple documents to answer complex questions (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). Unlike traditional question answering, which typically relies on information from a single passage, MHQA demands reasoning across disparate pieces of information, making it a more accurate reflection of human information-seeking behaviour (Chen et al., 2017). Despite recent advances in large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023), their ability to perform structured reasoning over multiple sources remains a challenging area, particularly when evidence must be gathered from diverse documents without explicit connections (Qi et al., 2019).

Existing approaches to MHQA typically follow a retrieve-then-read paradigm (Lewis et al., 2020; Karpukhin et al., 2020), where relevant documents are first retrieved based on the question, followed by a reading comprehension step to extract the answer. However, this sequential process often struggles with complex questions requiring multi-step reasoning, as the initial retrieval may fail to capture all necessary documents when relationships between different pieces of evidence are not explicitly considered [11]. Furthermore, most systems lack an effective mechanism to identify and address information gaps through iterative refinement (Trivedi et al., 2023). The increasing availability of powerful LLMs has opened new possibilities for MHQA, as these models demonstrate impressive reasoning capabilities (Wei et al., 2023; Wang et al., 2023). However, their application in multi-hop settings is often limited by several factors: (1) the inability to understand relationships between entities across different passages (Han et al., 2025), (2) the lack of structured representation of knowledge (Sun et al., 2024; Edge et al., 2025), and (3) the absence of systematic processes to identify and fill information gaps (He et al., 2024).

To address these limitations, we propose a novel Knowledge Graph-Enhanced Iterative Reasoning (KGEIR) framework for multi-hop question answering. Our approach combines the reasoning capabilities of LLMs with the structured representation of knowledge graphs, enabling more effective multi-hop reasoning through explicit modelling of entity relationships across documents. The key insight of our approach is that dynamically constructing and refining a knowledge graph during

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 134–143, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-014

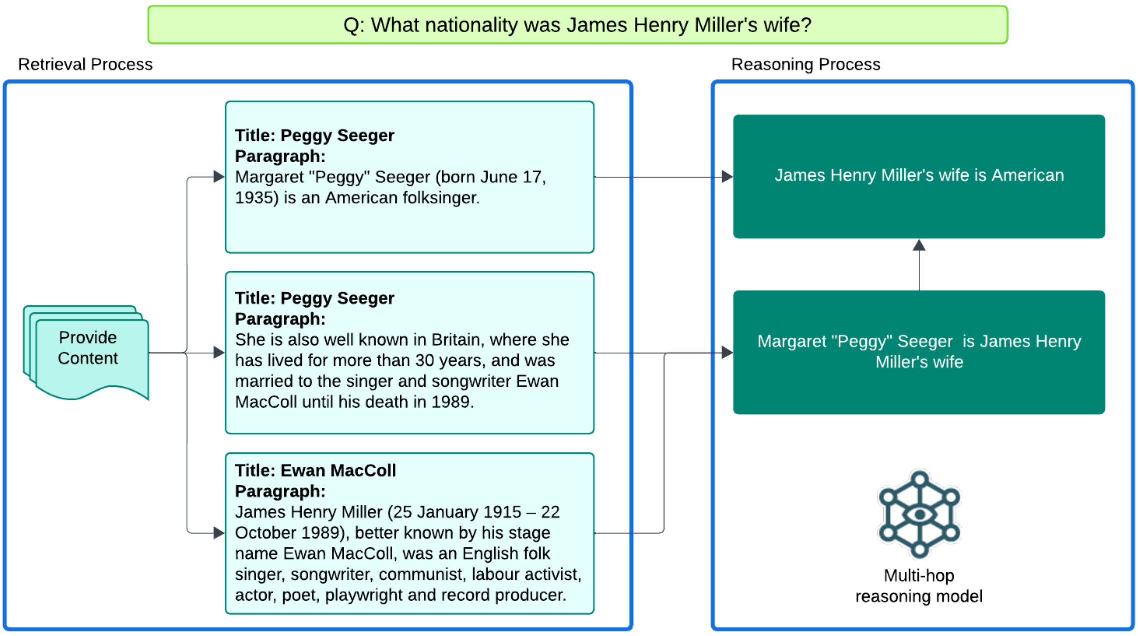


Figure 1: A common workflow on the MHQA task with an example from HotpotQA dataset. A regular MHQA question cannot get the answer from one single document but needs to retrieval multiply paragraphs from different documents. In here, the model firstly needs to retrieve the relevant paragraph across 2 different documents and identify that 'James Henry Miller' and 'Ewan MacColl' is one and the same person. Then it can make the connection between the fact that Peggy Seeger is his wife and the knowledge about her nationality (American).

the question-answering process provides an effective scaffold for reasoning, while also identifying information gaps that can guide targeted retrieval of additional context.

Our KGEIR framework operates through an iterative process: (1) initial retrieval of relevant paragraphs based on entities extracted from the question, (2) dynamic construction of a knowledge graph from retrieved paragraphs, (3) reasoning over the knowledge graph to attempt answering the question, (4) identification of information gaps in the knowledge graph, (5) targeted retrieval of additional paragraphs to fill these gaps, and (6) refinement of the knowledge graph and reasoning process. This iterative approach continues until sufficient information is gathered to answer the question confidently or a maximum number of iterations is reached.

We evaluate our approach on multiple multi-hop QA datasets, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022), demonstrating that KGEIR achieves significant improvements over strong baselines. Our analysis shows that the knowledge graph structure effectively guides the reasoning process of LLMs, while the iterative refinement process substantially improves answer accuracy by addressing information gaps identified during reasoning. The contributions of this paper are threefold:

1. A novel framework that leverages knowledge graphs to enhance multi-hop reasoning capabilities of large language models.

2. An iterative information-seeking approach that identifies and addresses knowledge gaps through targeted retrieval.

3. A comprehensive evaluation demonstrating the effectiveness of our approach on challenging multi-hop QA benchmarks.

Our results suggest that structuring information as explicit entity-relation graphs significantly enhances the multi-hop reasoning capabilities of LLMs, potentially opening new avenues for combining symbolic and neural approaches to complex question answering.

2 Related Works

This section examines recent advances in multi-hop question answering (MHQA), organised into complementary research directions that inform our KGEIR framework. By analysing the strengths and limitations of existing approaches, we demonstrate the need for our integrated framework.

2.1 Retrieval and Knowledge Structure Approaches

Recent retrieval methods for MHQA have progressed beyond simple matching to incorporate logical relevance and multi-hop connections. While dense retrievers like BGE (Xiao et al., 2024) perform well on single-hop tasks, they often struggle with capturing bridging information needed for complex reasoning. HopRAG (Liu et al., 2025) represents a significant advancement by introducing a logic-aware retrieval mechanism that connects passages through pseudo-queries and employs a retrieve-reason-prune paradigm. Their work demonstrated that indirectly relevant passages can serve as stepping stones to reach relevant ones, achieving notable results across multiple datasets. The approach, however, focuses primarily on enhancing retrieval rather than constructing explicit knowledge representations for reasoning.

Astute RAG (Wang et al., 2024) addresses imperfect retrieval by developing mechanisms to overcome knowledge conflicts and reasoning failures. They revealed that approximately 70% of retrieved passages do not directly contain true answers, highlighting the limitations of pure similarity-based retrieval. Similarly, BRIGHT (Su et al., 2025) demonstrates through their benchmark that even state-of-the-art retrievers struggle with multi-step reasoning tasks.

Knowledge structure approaches have emerged to provide explicit representations of entity relationships. G-Retriever (He et al., 2024) introduced a retrieval-augmented generation framework that enhances retrieval quality by leveraging graph structures to identify relevant information through entity-relation patterns. GraphRAG (Edge et al., 2025) builds hierarchical graph indices with knowledge graph construction and recursive summarisation, demonstrating the value of graph structures for organising complex information. Extract, Define, Canonicalise (Gutiérrez et al., 2025) presents an LLM-based framework for knowledge graph construction that systematically extracts entities and relations from text without extensive training or predefined schemata.

A key limitation across these approaches is their reliance on static construction processes and lack of explicit mechanisms to identify information gaps and iteratively refine knowledge representations, which our KGEIR framework specifically addresses.

2.2 Reasoning and LLM-Based Approaches

Recent reasoning approaches have increasingly leveraged structured representations to guide LLMs through complex multi-hop questions. Graph-based reasoning methods have shown particular promise in organising the reasoning process. Graph Elicitation (Park et al., 2024) decomposes multi-hop questions into sub-questions to form a graph and guides LLMs to answer based on the chronological order of the graph. Structure-Guided Prompting (Cheng et al., 2024) instructs LLMs in multi-step reasoning by exploring graph structures extracted from text. While effective, these approaches typically construct graphs as static scaffolds rather than dynamic structures that evolve through iterative refinement.

Graph Chain-of-Thought (Jin et al., 2024) augments LLMs by incorporating reasoning on graphs into the generation process, demonstrating that graph structures can significantly enhance LLMs' reasoning capabilities on tasks requiring structured knowledge. Reasoning with Graphs (Han et al., 2025) most closely aligns with our approach by structuring implicit knowledge into explicit graphs through multiple rounds of verification. Their results show significant improvements across logical reasoning and multi-hop question answering tasks, though their approach does not incorporate an iterative retrieval mechanism to address information gaps identified during reasoning.

The reasoning capabilities of LLMs have been extensively studied, revealing both strengths and limitations. Yang et al. (Yang et al., 2024) found that while models can connect information across sources, they benefit significantly from explicit guidance in complex scenarios, particularly as reasoning hops increase. Huang et al. (Huang et al., 2024) demonstrated that even advanced LLMs struggle to identify and correct errors in their reasoning without external guidance, underscoring the importance of providing explicit structures to guide the reasoning process.

Various approaches have been proposed to enhance LLMs' reasoning. Self-RAG (Asai et al., 2023) introduced a framework for retrieval, generation, and critique through self-reflection, while REFEED (Yu et al., 2023) employs a multi-round retrieval-generation framework using feedback to refine retrieval steps. SAFE-RAG (Liang et al., 2025) highlighted the importance of reliable reasoning over retrieved information, showing that without proper verification mechanisms, LLMs can produce inconsistent responses. However, these approaches typically lack explicit mechanisms to identify specific information gaps or leverage structured representations for reasoning.

2.3 Iterative Refinement Approaches

Iterative approaches to information retrieval and reasoning have gained significant traction, addressing multi-hop question answering challenges through progressive refinement. When compared with tree-structured RAG approaches like RAPTOR (Sarthi et al., 2024) and SiReRAG (Zhang et al., 2025), graph-structured approaches, such as HopRAG (Liu et al., 2025) demonstrate superior performance by enabling flexible logical modelling, cross-document organisation, and efficient construction.

The HippoRAG framework (Yang et al., 2024) introduces a neurobiologically inspired approach to long-term memory for LLMs, implementing a system that prioritises relevance signals and iteratively refines its understanding. However, their approach does not explicitly model the graph evolution process or use graph structures to identify information gaps. ActiveRetrieval (Jiang et al., 2023) actively queries a corpus during the generation process, using intermediate reasoning states to guide retrieval. This approach demonstrates the value of dynamically adjusting retrieval based on the current reasoning state, a principle that our KGEIR framework incorporates through gap-aware retrieval.

While these existing approaches have made significant strides in different aspects of the MHQA challenge, they typically address only part of the problem. KGEIR uniquely integrates dynamic knowledge graph construction, gap identification, and iterative refinement into a unified framework that addresses the full spectrum of challenges in multi-hop question answering, differentiating it from existing approaches that typically address only part of the problem.

3 Methods

We introduce KGEIR (Knowledge Graph-Enhanced Iterative Reasoning), a novel framework for multi-hop question answering that combines the reasoning capabilities of large language models with the structural advantages of knowledge graphs. This section describes our approach, which dynamically constructs and refines knowledge graphs to support iterative reasoning over multiple documents.

3.1 Problem Analysis

Multi-hop question answering requires integrating information across multiple sources to derive answers that cannot be found in any individual source. We formalize this task as follows: Given a question q and a corpus of documents $\mathcal{D} = \{d_1, d_2, ..., d_N\}$, MHQA aims to produce an answer a by reasoning over a subset of supporting documents $\mathcal{S} \subset \mathcal{D}$ where:

- No single document $d_i \in \mathcal{S}$ contains sufficient information to answer q.
- The answer a requires establishing relationships between information in different documents.
- The reasoning process can be represented as a sequence of hops between documents, forming a path:

$$d_{i_1} \to d_{i_2} \to ... \to d_{i_k} \to a$$

Traditional approaches follow a retrieve-then-read paradigm that can be formalised as:

$$a = \mathcal{R}(\mathcal{T}(q, \mathcal{D}), q)$$

Where $\mathcal{T}$ is a retrieval function that selects relevant documents, and$\mathcal{R}$ is a reading function that extracts the answer. This approach faces challenges with multi-hop questions as the initial retrieval often fails to capture all necessary information.

Our KGEIR framework reformulates this problem by introducing an iterative graph-based approach:

$$a = \mathcal{R}(G_k, q)$$

Where G_k is a knowledge graph constructed and refined through k iterations of retrieval and reasoning. Each iteration identifies information gaps and retrieves additional context to fill these gaps, progressively enriching the graph until sufficient information is gathered to answer the question.

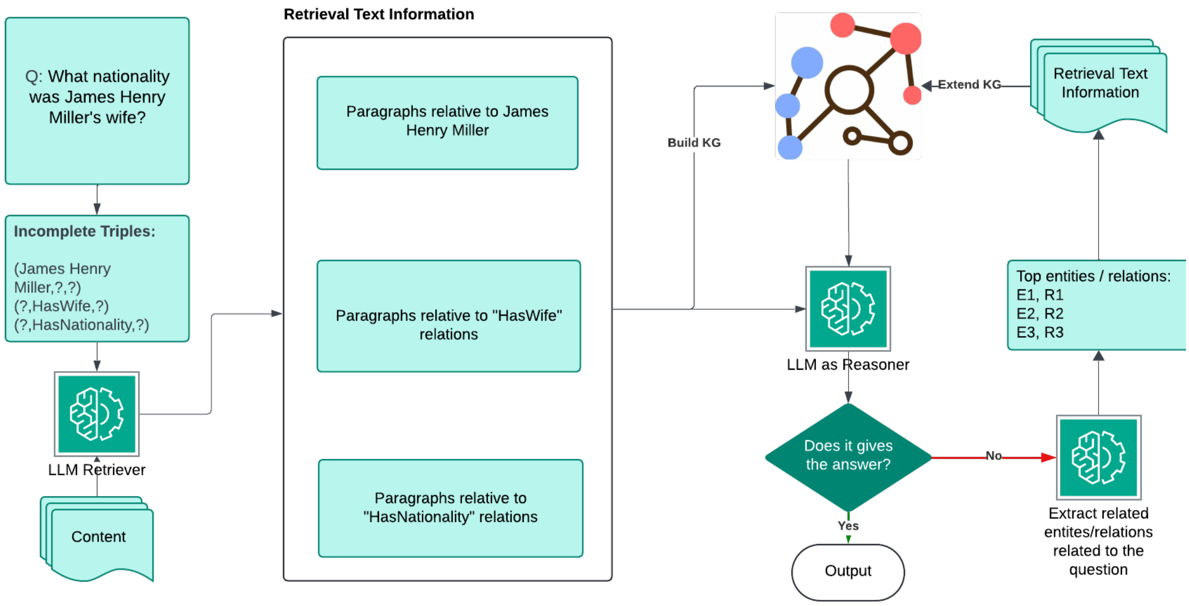


Figure 2: KGEIR framework workflow for multi-hop QA, illustrated with the example "What nationality was James Henry Miller's wife?" The process begins with extracting incomplete triples from the question (left), followed by multi-faceted retrieval extracting paragraphs relevant to the main entity and relations. The initial knowledge graph is constructed from retrieved paragraphs (centre), enabling structured reasoning by the LLM. If the current graph lacks sufficient information, the system identifies missing entities and relations to guide targeted retrieval (right), iteratively refining the knowledge graph until a confident answer can be produced. This dynamic enhancement process addresses the limitations of static retrieval approaches by adaptively exploring the information space based on reasoning requirements.

3.2 Initial Knowledge Graph Construction

The first component of KGEIR is constructing an initial knowledge graph from the question and corpus. As shown in Figure 1, this process involves question analysis, initial retrieval, and graph formation. Given a question, we first identify key entities and potential relationships required to answer it. For the question "What nationality was James Henry Miller's wife?" (Figure 1), we extract incomplete triples: (James Henry Miller, ?, ?), (?, HasWife, ?), and (?, HasNationality, ?). These incomplete triples capture both explicit entities mentioned in the question and implicit relations necessary to answer it. We only crop the corpus if it reaches the limitation of LLM's context length. Then, combining the text corpus with these extracted entities and relations, we design a prompt for the LLM to retrieve the relevant paragraphs from the corpus. This multi-faceted retrieval approach targets documents containing information about the entities ("James Henry Miller"), relations ("HasWife"), and properties ("HasNationality") in the query. This strategy ensures a broader coverage than traditional retrieval methods that focus on entities only. All our prompts used throughout the paper are available upon request.

From the retrieved paragraphs, we extract entities and relationships to construct the initial knowledge graph. This process creates a structured representation of the information contained in the retrieved documents, converting unstructured text into an explicit entity-relation graph. While valuable, this initial graph often lacks crucial connections needed to answer complex multi-hop questions, necessitating our dynamic enhancement approach.

3.3 Dynamic Knowledge Graph Enhancement

The core innovation of KGEIR is its dynamic approach to enhancing the knowledge graph through targeted retrieval and iterative refinement, as shown in the right portion of Figure 1. After constructing the initial graph, an LLM reasoner attempts to answer the question using the available knowledge structure. Upon determining that the current graph does not contain sufficient information for a confident answer to the question, the system initiates an enhancement cycle that operates through a systematic information-seeking paradigm.

The enhancement process begins with a gap analysis mechanism that examines the knowledge graph structure to identify missing entities and relations. This mechanism employs specialised prompting techniques to direct the LLM's attention to specific structural deficiencies in the current graph. As illustrated in Figure 1, the system identifies entities that would be most informative for answering the query, modelling information necessity rather than merely information relevance. Following gap identification, the system employs relation-aware retrieval to efficiently locate documents containing the missing information. This targeted retrieval strategy differs significantly from traditional similarity-based approaches by formulating queries specifically designed to bridge identified knowledge gaps. The retrieval component employs both entity-centric and relation-centric query formulation to ensure comprehensive coverage of the missing information. The retrieved information undergoes structured extraction and integration into the existing knowledge graph through our graph extension mechanism (labelled "Extend KG" in Figure 1). This integration process preserves existing graph structure while incorporating new entities and relations, creating a progressively more comprehensive knowledge representation with each iteration. The enhancement cycle continues iteratively, with each cycle refining the knowledge graph until it contains sufficient information for confident reasoning or reaches a predetermined iteration limit. This dynamic refinement process enables KGEIR to overcome the limitations of static retrieval approaches, adaptively exploring the information space as directed by reasoning requirements rather than surface-level query similarity.

3.4 Knowledge-Guided Reasoning and Assessment

The final component of KGEIR leverages the dynamically enhanced knowledge graph to perform multi-hop reasoning and answer the question. Unlike approaches that reason directly over retrieved text, KGEIR reasons over the structured entity-relation graph, allowing for more precise navigation through complex information. The reasoning process leverages both the graph structure and the original retrieved passages, combining the advantages of structured knowledge representation with the contextual richness of the original text. As illustrated in Figure 1, the LLM reasoner identifies relevant paths through the knowledge graph that connect question entities to potential answers. For our example question, the reasoner would identify paths connecting "James Henry Miller" through the "HasWife" relation to his spouse, and then through the "HasNationality" relation to the target answer. By traversing these explicit relationship paths, the system effectively performs multi-hop reasoning while maintaining clarity about the evidence supporting each hop.

The reasoning process includes a simple verification step (the decision node in Figure 1) where the LLM determines if the current graph provides direct supporting information to answer the question. If more information is needed, the system triggers another enhancement cycle; otherwise, it proceeds to generate the final answer.

4 Experiment

4.1 Setup

Dataset and Retrieval Parameter For comprehensive evaluation of KGEIR, we conducted experiments on three established multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2023). These datasets represent varying degrees of reasoning complexity, including 2-hop, 3-hop, and 4-hop inference chains. Following established evaluation practices in this domain (Zhang et al., 2025), we selected a sample of 1,000 questions from each dataset's validation set. For the hyperparameter setting, we set the number of retrieved paragraphs to five for each iteration of the enhancement cycle, for a maximum of three iterations. If the model still cannot find the answer at this point, the question is marked as failed.

Baselines We evaluated KGEIR against representative methods spanning different approaches to multi-hop reasoning. We included both sparse retrieval with BM25 (Robertson and Zaragoza, 2009) and dense retrieval with BGE (Xiao et al., 2024) to establish performance baselines for non-structured approaches. We compared against the published results in Liu et al. (2025) on leading tree-structured systems, including RAPTOR (Sarthi et al., 2024) and SiReRAG (Zhang et al., 2025), as well as graph-based approaches namely GraphRAG (Edge et al., 2025) and HippoRAG (Gutiérrez et al., 2025). For all structured systems, GPT-4o has been used to maintain consistency between implementations.

Evaluation Metric We assessed performance using exact match (EM) and F1 scores as same as the setting in HopRAG model (Liu et al., 2025), which measures the precision of answer generation at different granularities. The EM metric requires exact correspondence with reference answers, while F1 combines precision and recall at the token level to provide a more nuanced measure of partial correctness. We focused exclusively on answer quality metrics rather than retrieval metrics, as several baseline systems generate synthetic content (such as summaries) that would make direct retrieval comparison inequitable.

4.2 Result Analysis

Table 1 presents a comprehensive evaluation of our proposed KGEIR framework against established baselines across three multi-hop QA datasets. Thus, KGEIR achieves competitive results with state-of-the-art methods, our novel mechanisms of dynamic knowledge graph construction and iterative reasoning.

On MuSiQue, KGEIR achieves 44.50% EM (exact matches) and 53.12% F1, showing modest improvements over HopRAG (42.20% EM, 54.90% F1). For HotpotQA, we observe performance of 63.15% EM and 76.77% F1, slightly higher than HopRAG's 62.00% EM and 76.06% F1. On 2WikiMultiHopQA, our approach achieves 59.13% EM and 69.55% F1, which is competitive though slightly lower than HopRAG's 61.10% EM and 68.26% F1. These results demonstrate that KGEIR achieves comparable performance to the current state-of-the-art while introducing a fundamentally different approach to multi-hop reasoning. The primary contribution of KGEIR is not a significant leap in raw performance metrics, but rather the introduction of a novel framework that enhances the reasoning process through explicit knowledge modelling and iterative refinement.

In terms of approach, KGEIR differs from HopRAG in several key aspects. While HopRAG prioritises logical connectivity between passages through pseudo-queries and multi-hop traversal, KGEIR focuses on dynamically constructing and refining explicit knowledge representations. Unlike HopRAG, which integrates similarity with logical relations when constructing edges, KGEIR explicitly models information gaps and uses these to guide targeted retrieval. The performance comparisons with traditional retrievers (BM25: 31.77% avg. EM, BGE: 36.17% avg. EM) highlight the significant advantages of structured approaches. Meanwhile, GraphRAG's lower performance (22.10% avg. EM) suggests that static knowledge graph construction alone is insufficient without iterative refinement mechanisms. Similar to how HopRAG positioned itself against SiReRAG by emphasising its streamlined graph structure without additional summary nodes, KGEIR introduces a novel dynamic knowledge graph construction process that evolves throughout reasoning. Our approach does not require pre-constructed knowledge graphs or complex graph preprocessing, instead, it builds and refines graph representations as reasoning progresses.

4.3 Ablation Experiment and Discussion

To evaluate the effectiveness of KG-based reasoning in our framework, we performed an ablation study comparing different reasoning methods following the retrieval phase. We examined four distinct approaches: (1) Vanilla (direct LLM reasoning without prompting), (2) Chain-of-Thought (CoT) (Wei et al., 2023), (3) Tree-of-Thought (ToT) (Yao et al., 2023), and (4) our complete KGEIR approach with knowledge graph reasoning. For all experiments, we used Gemma3-27B as the base model and maintained consistent dataset settings with our main evaluation. Performance was measured based on semantic correctness relative to ground truth answers.

As shown in Table 2, KGEIR consistently outperforms all baseline reasoning methods across all datasets, achieving an average improvement of 2.26% over ToT. The performance improvement is particularly pronounced on the HotpotQA dataset, where KGEIR achieves 62.20% accuracy compared to 57.70% for ToT—a 4.50% absolute improvement. All results suggest that our knowledge graph approach is very effective for complex bridging questions that require connecting information across multiple documents.

Table 2 shows a clear progressive improvement pattern (Vanilla → CoT → ToT → KGEIR), demonstrating the value of increasingly structured reasoning approaches. While CoT provides modest gains over vanilla reasoning (48.11% vs. 48.47% on average), ToT's tree-structured exploration offers more substantial improvements (53.85%). However, KGEIR's explicit modelling of entity relationships through dynamic knowledge graphs

Table 1: Comparison of RAG methods across datasets with baseline results from the cited literature.

Method	MuSiQue		2WikiQA		HotpotQA		Average	
	EM [%]	F1 [%]	EM [%]	F1 [%]	EM [%]	F1 [%]	EM [%]	F1 [%]
BM25	13.80	21.50	40.30	44.83	41.20	53.23	31.77	39.85
BGE	20.80	30.10	40.10	44.96	47.60	60.36	36.17	45.14
GraphRAG	12.10	20.22	22.50	27.49	31.70	42.74	22.10	30.15
RAPTOR	36.40	49.09	53.80	61.45	58.00	73.08	49.40	61.21
SiReRAG	40.50	53.08	59.60	67.94	61.70	**76.48**	53.93	65.83
HopRAG	42.20	**54.90**	**61.10**	68.26	62.00	76.06	55.10	**66.40**
KGEIR	**44.50**	53.12	59.13	**69.55**	**63.15**	73.77	**55.59**	65.48

Table 2: Comparison of ablation study between different reasoning methods across datasets.

Method	MuSiQue	2WikiQA	HotpotQA	Average
Vanilla (LLM 'as is')	42.60	62.21	40.62	48.47
CoT (LLM with CoT prompt)	44.50	57.25	42.57	48.11
ToT (LLM with ToT prompt)	45.65	58.21	57.70	53.85
KGEIR	46.45	59.69	62.20	56.11

provides the most effective reasoning framework (56.11%).

Interestingly, on 2WikiQA, the performance gap between reasoning methods is less pronounced, with vanilla LLM reasoning achieving a surprisingly high 62.21%. This suggests that for certain types of questions, the base reasoning capabilities of modern LLMs may be sufficient when retrieving appropriate context. Nevertheless, KGEIR still provides the most consistent performance across all datasets, demonstrating the robustness of our approach to different question types and reasoning complexities.

These results validate our hypothesis that structuring multi-hop reasoning through explicit knowledge graphs enhances LLMs' ability to connect information across documents, particularly for complex questions requiring multiple reasoning steps. The dynamic construction and refinement of knowledge representations provide a more interpretable and effective reasoning process compared to traditional prompting methods.

5 Conclusion

In this paper, we presented KGEIR, a novel framework that enhances multi-hop question answering through dynamic knowledge graph construction and iterative refinement. Unlike traditional retrieve-then-read paradigms, KGEIR explicitly models entity relationships across documents and systematically identifies information gaps to guide targeted retrieval. This iterative knowledge refinement process provides both a structured scaffold for LLM reasoning and an effective mechanism to address the inherent limitations of similarity-based retrieval for complex questions.

Our comprehensive evaluation across three multi-hop QA benchmarks demonstrates KGEIR's effectiveness, achieving competitive or superior performance compared to state-of-the-art methods. The most significant improvements appear on complex bridging questions, confirming our approach's strength in scenarios requiring cross-document reasoning. Ablation experiments reveal that structured knowledge graph reasoning consistently outperforms traditional prompting methods, with our full KGEIR model providing absolute improvements of up to 4.50% over Tree-of-Thought prompting.

The integration of dynamic knowledge graph construction with iterative reasoning represents a promising direction for addressing complex information needs in NLP systems. By bridging symbolic and neural approaches, KGEIR offers a principled solution to the challenges of information fragmentation and implicit relationships that characterise multi-hop reasoning tasks. We may extend this framework to incorporate uncertainty handling and conflicting information resolution, potentially expanding its applicability to a broader range of knowledge-intensive applications beyond question-answering.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. ArXiv:2310.11511 [cs].

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.

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, Vancouver, Canada. Association for Computational Linguistics.

Kewei Cheng, Nesreen K. Ahmed, Theodore Willke, and Yizhou Sun. 2024. Structure Guided Prompt: Instructing Large Language Model in Multi-Step Reasoning by Exploring Graph Structure of the Text. ArXiv:2402.13415 [cs].

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. ArXiv:2404.16130 [cs].

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2025. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. ArXiv:2405.14831 [cs].

Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Hui Liu, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and Jiliang Tang. 2025. Reasoning with Graphs: Structuring Implicit Knowledge to Enhance LLMs Reasoning. ArXiv:2501.07845 [cs].

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. ArXiv:2402.07630 [cs].

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large Language Models Cannot Self-Correct Reasoning Yet. ArXiv:2310.01798 [cs].

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. ArXiv:2305.06983 [cs].

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 163–184, Bangkok, Thailand. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 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.

Xun Liang, Simin Niu, Zhiyu Li, Sensen Zhang, Hanyu Wang, Feiyu Xiong, Jason Zhaoxin Fan, Bo Tang, Shichao Song, Mengwei Wang, and Jiawei Yang. 2025. SafeRAG: Benchmarking Security in Retrieval-Augmented Generation of Large Language Model. ArXiv:2501.18636 [cs].

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation. ArXiv:2502.12442 [cs].

Jinyoung Park, Ameen Patel, Omar Zia Khan, Hyunwoo J. Kim, and Joo-Kyung Kim. 2024. Graph Elicitation for Guiding Multi-Step Reasoning in Large Language Models. ArXiv:2311.09762 [cs].

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering Complex Open-domain Questions Through Iterative Query

Generation. 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 2590–2602, Hong Kong, China. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. ArXiv:2401.18059 [cs].

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. ArXiv:2407.12883 [cs].

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. ArXiv:2307.07697 [cs].

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].

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554. Place: Cambridge, MA Publisher: MIT Press.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. 2024. Astute RAG: Overcoming Imperfect Retrieval Augmentation and Knowledge Conflicts for Large Language Models. ArXiv:2410.07176 [cs].

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv:2203.11171 [cs].

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. ArXiv:2201.11903 [cs].

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. ArXiv:2309.07597 [cs].

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do Large Language Models Latently Perform Multi-Hop Reasoning? ArXiv:2402.16837 [cs].

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. ArXiv:2305.10601 [cs].

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving Language Models via Plug-and-Play Retrieval Feedback. ArXiv:2305.14002 [cs].

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. 2025. SiReRAG: Indexing Similar and Related Information for Multihop Reasoning. ArXiv:2412.06206 [cs].

From Handcrafted Features to LLMs: A Comparative Study in Native Language Identification

Aliyah C. Vanterpool
Montclair State University
New Jersey, USA
aliyahvan@gmail.com

Katsiaryna Aharodnik
CUNY Graduate Center
New York, USA
kaharodnik@gradcenter.cuny.edu

Abstract

This study compares a traditional machine learning feature-engineering approach to a large language models (LLMs) fine-tuning method for Native Language Identification (NLI). We explored the COREFL corpus, which consists of L2 English narratives produced by Spanish and German L1 speakers with lower-advanced English proficiency (C1) (Lozano et al., 2020). For the feature-engineering approach, we extracted language productivity, linguistic diversity, and n-gram features for Support Vector Machine (SVM) classification. We also looked at sentence embeddings with SVM and logistic regression. For the LLM approach, we evaluated BERT-like models and GPT-4. The feature-engineering approach, particularly n-grams, outperformed the LLMs. Sentence-BERT embeddings with SVM achieved the second-highest accuracy (93%), while GPT-4 reached an average accuracy of 90.4% across three runs when prompted with labels. These findings suggest that feature engineering remains a robust method for NLI, especially for smaller datasets with subtle linguistic differences between classes. This study contributes to the comparative analysis of traditional machine learning and transformer-based LLMs, highlighting current LLM limitations in handling domain-specific data and their need for larger training resources.

1 Introduction

The role of a learner's native language (L1) in second language (L2) acquisition has been widely addressed in second language acquisition (SLA) literature (Lado, 1957; Corder, 1975). SLA research has shown that the spelling, grammar, and lexicon used in L2 writing are often influenced by patterns and rules from a learner's L1. However, the extent of L1 impact on L2 performance remains difficult to determine precisely.

With the emergence of learner corpora, it has become possible to empirically test SLA hypotheses and explore how different L1s manifest in L2 writing. One application of this is the Native Language Identification (NLI) task, which uses automated methods to predict a learner's L1 based on their L2 writing. Prior studies have demonstrated high performance for feature-engineered machine learning (ML) approaches to NLI. However, research examining the applicability of large language models (LLMs) to NLI remains limited. Moreover, there is a lack of studies directly comparing LLMs with traditional feature-engineered pipelines within the same experimental paradigm.

The current study addresses this gap by comparing a traditional feature-engineering ML approach to transformer-based LLMs for the NLI task. As a secondary goal, we explore both methods using a relatively small but unique learner corpus composed of video-based written narratives. This corpus offers more structured and homogeneous data than the topic-based essays commonly used in prior NLI studies. We report the results of both NLI approaches and discuss their implications for SLA research.

This paper is structured as follows: Section 2 introduces previous research. Section 3 outlines the methodology, including a description of the COREFL corpus and training/testing techniques. Section 4 presents the results of both approaches. Section 5 discusses the findings and implications for SLA and NLI. Section 6 provides the conclusion and suggests future research directions.

2 Related Work

In NLI research, findings are often interpreted through the lens of Second Language Acquisition (SLA) and linguistic transfer. Several theoretical approaches from SLA have served as a founda-

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 144–153, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-015

tion for this task. One of the most influential is the Contrastive Analysis Hypothesis (CAH; Lado, 1957), which posits that difficulties in second language learning arise from differences between the learner's first language (L1) and the target language (L2). Language typology plays a key role in this process: the more similar two languages are, the more likely learners are to experience positive transfer that facilitates acquisition; conversely, typologically distant languages tend to result in more negative transfer and errors.

Linguistic transfer refers to the application of phonological, morphological, syntactic, or lexical rules from one language to another (Odlin, 1989). For instance, a native speaker of a pro-drop language, such as Spanish, may incorrectly omit subjects when constructing sentences in a non-pro-drop language like German. The likelihood and nature of transfer errors depend not only on structural differences between the languages but also on the learner's level of proficiency (Montrul, 2014). As learners become more proficient in the L2, they tend to make fewer transfer-based errors.

In the context of NLI, the underlying assumption is that classification algorithms can detect subtle linguistic patterns in learners' L2 that reflect L1 influence, such as deviations in syntactic structure, part-of-speech usage, or inconsistencies in lexicon and use these cues to identify the writer's native language. These linguistic traces provide support for theoretical approaches in SLA and help explore the phenomenon of cross-linguistic influence or transfer (Jarvis and Crossley, 2012; Tsur and Rappoport, 2007).

Prior studies have consistently demonstrated the effectiveness of n-gram-based features for NLI. For instance, several studies have found character n-grams to be among the most discriminative features (Koppel et al., 2005; Markov et al., 2022), while others have reported high classification performance using lexical and part-of-speech (POS) n-grams. Jarvis et al. (2013), for example, achieved an accuracy of 83.6% using word n-grams, and Markov et al. (2022) reported accuracies ranging from 80% to 90% using character n-grams with high values of n (up to n=9). Furthermore, combinations of POS n-grams and error features have yielded precision and recall scores exceeding 80% (Aharodnik et al., 2013; Kochmar, 2011). For example, Kochmar (2011) reported 84% accuracy using a combined feature set of character n-grams, POS n-grams, and corpus-derived error rates for classifying Romance and Germanic languages. In contrast, fewer NLI studies have examined features that reflected language productivity and lexical diversity, such as function word and content word ratios, mean length of utterance in words, and type-token ratio. However, these features may also be informative, as learners may exhibit L1-influenced lexical and syntactic patterns in their writing. For example, some studies emphasized that function words have contributed to high-performing models when combined with n-grams and error features (Koppel et al., 2005; Wong and Dras, 2009).

Studies exploring NLI with LLMs have yielded mixed results. For example, Lotfi et al. (2020) reported an accuracy of 89% on the test set for TOEFL11 and 94.2% on 5-fold cross validation for ICLE Corpus using GPT-2. These results indicated that the open source GPT model (GPT-2) was higher than the traditional machine learning approaches, with the best performing model achieving 88.2% accuracy with the SVM (Malmasi et al., 2017). However, studies have shown lower performance for BERT-like LLMs compared to a GPT-2 model. For example, 80.8% accuracy was attained using BERT-base-uncased when tested on the TOEFL11 corpus test set (Lotfi et al., 2020). Importantly, few studies have directly compared traditional machine learning and LLM-based approaches within the same experimental framework.

Moreover, LLM performance appears to be sensitive to dataset size. For instance, Steinbakken and Gambäck (2020) found that BERT-based models reached 85.3% accuracy on the TOEFL11 dataset, but accuracy improved to 90.2% when using the larger Reddit-L2 dataset. These findings suggest that LLMs require larger and more diverse data to perform optimally, highlighting the need for further research that examines LLM effectiveness on datasets of varying sizes and content types.

The nature of the data itself also plays a critical role in classification performance. Most NLI studies have relied on the TOEFL11 corpus, which contains argumentative essays on various topics (Malmasi and Dras, 2015). While high performance has consistently been reported for this dataset (Koppel et al., 2005; Malmasi and Cahill, 2015), its topic-based structure introduces the risk of content bias, particularly when using content-sensitive features such as word and character n-grams. Studies on cross-corpora evaluation have found that

Features	Description
Narrative Microstructure	
MLU(w)	Mean Length of Utterance in Words: ratio of total word tokens to total number of sentences per text
FCR	Function-to-Content Word Ratio. *Function words:* auxilliaries, pronouns, determiners, prepositions, conjunctions, particles. *Content words:* nouns, verbs, adjectives, adverbs.
TTR	Type-Token Ratio: ratio of unique words to total words.
POS_fc	Part-of-speech frequency counts (e.g., the number of NOUNs, VERBs, ADJs, etc. per text).
N-gram Features	
POS n-grams	Sequences of POS tags (e.g., ”DET NOUN”, ”NOUN VERB ADV”).
Word n-grams	Sequences of words (e.g., ”he walked”, ”the baby ate”)
Character n-grams	Sequences of characters (e.g., ”ing”, ”ys”, ”ies”)

Table 1: Overview of linguistic productivity, language diversity, and n-gram features included in the study.

genre-diverse corpora produce a higher accuracy when tested on a genre-specific corpus than the reverse. However, overall accuracy remains relatively low, as many features useful for NLI are genre-dependent (Malmasi and Dras, 2015). More structured datasets, such as those based on picture- or video-based narratives, can be used as an alternative for a more consistent feature extraction across participants.

The current study addresses these gaps by comparing a traditional feature-engineering approach with supervised machine learning classifiers and the fine-tuning of LLMs within a single experimental setup. We examine both previously validated feature sets, such as n-grams, and a complementary set of language productivity and diversity measures. This approach aims to assess whether these additional linguistic features enhance classification performance and provide deeper insights into L1-specific patterns in learner writing. To minimize topic-related bias in L1 identification, we apply both methods to a more homogeneous dataset.

3 Methods

3.1 Dataset

We used the COREFL corpus (Lozano et al., 2020). The corpus contained English L2 learner data of Spanish and German L1 backgrounds. Only learners with a lower advanced level of English proficiency (C1) were included in the study. The writers' age ranged from 18 to 60 years old. The data consisted of 84 German and 79 Spanish files with one file per participant. The participants watched a 4-minute video clip about Charlie Chaplin and summarized the story in a written essay.

Language	Total Files	VS	BDS
German	84	13	7
Spanish	79	17	7
Total	163	30	14

Table 2: Total number of files and the number of files used for validation and blind test sets for both language groups. VS = Validation Set. BDS = Blind Dataset.

3.2 Feature-Engineering Approach

The feature-engineering step focused on selecting and automatically extracting specific features that best characterized the data. The features for this study included two sets described in detail in Table 1.

The pre-processing step involved data cleaning and feature extraction. Data cleaning consisted of basic steps: removing special characters, removing punctuation, lowercasing, tokenization, and POS tagging. All features were extracted from narratives using bash scripts. The POS tagging was implemented using *en_core_web_trf* with Spacy Python package. All bash scripts and Python code is available on GitHub[1]:

The extracted features were used as input for supervised machine learning binary classification. We implemented the Support Vector Machine clas-

[1]https://github.com/AliyahVanterpool/ml_features_vs_llm.git

Testing Set	Feature	Accuracy	F1
VS	All	0.70	0.70
BDS	All	**0.79**	**0.78**
VS	MTF	0.67	0.64
BDS	MTF	0.64	0.64
VS	POS_fc	0.63	0.63
BDS	POS_fc	0.71	0.71
VS	MLU(w)	0.67	0.62
BDS	MLU(w)	0.50	0.48
VS	TTR	0.60	0.57
BDS	TTR	0.50	0.33
VS	FCR	0.60	0.55
BDS	FCR	**0.79**	**0.78**

Table 3: Highest accuracy and F1 for language productivity and diversity features. VS = Validation Set. BDS = Blind Dataset. All = MLU(w), TTR, FCR, POS_fc. MTF = MLU(w), TTR, FCR.

sifier (SVM; Cortes and Vapnik, 1995) with linear and rbf kernels, Logistic Regression (Cox, 1958; Hosmer Jr et al., 2013), and K-Nearest Neighbors (KNNs) classifier (Cover and Hart, 1967). We compared the performance across feature sets and classifiers. Table 2 shows the total number of files and those allocated to the validation and blind test sets for both language groups. The following models were included in the analysis: 1) all features (ALL = MLU(w) + TTR + FCR + POS_fc); 2) each feature from ALL models individually; 3) MTF feature set (MLU(w) + TTR + FCR); 4) word n-grams (bigrams, trigrams); 5) POS n-grams (bigrams, trigrams); and 6) character n-grams (four-grams to nine-grams).

The training dataset was created using 90% of the entire dataset, while 10% was held out for the blind test set. 80% of the training data was used for training and 20% for validation. The classifier training and parameter tuning was implemented using scikit-learn package in Python (Pedregosa et al., 2011). The kernels and c-parameter were explored to evaluate which models performed the best.

We also looked at Sentence-BERT embeddings (Reimers and Gurevych, 2019). We implemented *all-MiniLM-L6-v2*, a distilled BERT-based model from the Sentence Transformers. These embeddings were used as feature vectors for downstream binary ML classification with SVM and Logistic Regression. We evaluated the performance of both classifiers and reported the accuracy for the blind test set.

3.3 LLM Approach

For the LLM approach, we explored BERT-like models (Devlin et al., 2019). These models were ALBERT (Lan et al., 2019), BERT-base-multilingual-cased, BERT-base-uncased, DistilRoBERTa-base, DistilBERT-base-uncased, and XLM-RoBERTa-base. We fine-tuned these pre-trained models for sequence classification using the learner corpus. The fine-tuning process involved training each model on 80% of the entire dataset, with 20% validation for a maximum of 3 epochs with a learning rate of 1e-5 and a batch size of 8. We experimented with frozen layers, however the models with all layers demonstrated better results and thus were reported in our study.

Additionally, we evaluated GPT-4 performance across three runs in two ways - 1) when tested on the blind dataset with class labels provided; and 2) no labels given. When GPT-4 was provided with labeled data, the prompt was: *The following English text is written by either a native German speaker or native Spanish speaker. What is the native language of the writer of this text: German or Spanish? Explain your choice in 1-2 sentences.* The prompt for unlabeled data was: *The following English text is written by a non-native speaker. What is the native language of the writer of this text? Explain your choice in 1-2 sentences.*

3.4 Testing and Evaluation Metrics

For the feature-engineering approach, we used three testing techniques: validation set, blind dataset, and k-fold cross validation (CV). The validation split was 20% of the training dataset. The blind dataset consisted of 10% of the entire dataset held out for testing and not included in the training. The blind dataset included 7 random files for each label (14 files in total). For K-fold CV, k ranged from 5-10 and the best k (k = 7) was reported. We reported the results for the SVM classifier since it demonstrated the best performance. We evaluated the best accuracy for linear and rbf kernels, and for C-parameter value. We also calculated feature importance scores with Random Forest Classifier for word bigrams and trigrams from the blind test set to identify those n-grams that impacted the classifier's decisions.

For the LLMs approach, we looked at both the validation and blind dataset results and reported the blind test results. Cross-validation techniques was computationally expensive for the BERT-like models, hence those were not reported for this study.

4 Results

4.1 Feature Engineering Approach

For the feature engineering approach, the best performing model was the model with all productivity and diversity features combined (ALL; 79% accuracy and 78% F1-score). K-fold CV for all feature models produced the highest mean accuracy of 72.5%. The productivity and diversity measures are described in Table 3. Models with individual features showed the highest accuracy (79%) and F1-score (78%) for function-to-content ratios with k-fold CV at 57.7%.

Among n-gram features, word bigrams and trigrams as well as character four- and five-grams attained the highest accuracy and F1 for both the validation and blind datasets. These results are shown in Table 4. The highest accuracy of 100% (95% CI [0.78, 1.00], Wilson interval) was achieved by word bigrams when tested on the blind dataset. The k-fold CV accuracy with k = 7 was 90.6% for word bigrams. The best models for the validation set were word bigrams and trigrams, as they acquired an accuracy of 93% (95% CI [0.79, 0.98], Wilson interval). The k-fold CV accuracy for trigrams was 91.3%. POS bigrams had the highest accuracy when tested on the validation set (87%; 95% CI [0.70, 0.95], Wilson interval) and POS trigrams acquired the highest accuracy of 79% (95% CI [0.52, 0.92], Wilson interval) when tested on the blind dataset. The K-fold CV accuracy was 81.2% for POS bigrams and 82.5% for POS trigrams. Overall, the results for n-gram features demonstrated the highest accuracy and stable results across different testing techniques (validation, blind test, and K-fold CV).

The highest accuracy for sentence embeddings with SVM was 93% and 78% with logistic regression when tested on the blind dataset. Additionally, the SVM embedding results performed better than the language productivity and diversity measures. However, sentence embeddings results were lower than the word bigram results. The best performing models for the feature-engineering approach, including sentence embeddings, are displayed in Figure 1.

4.2 LLM Approach

The LLM approach was separated into two parts: (1) BERT-like models with a classification layer, and (2) GPT-4 results. For the first part, we reported the performance of the blind test set. For the second part, we provided the average GPT-4 results across three runs for prompting with and without labels.

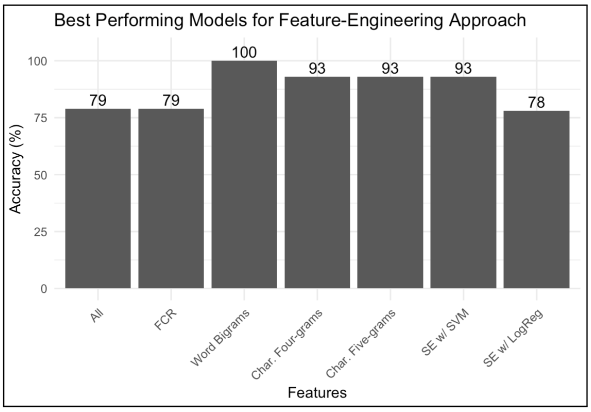


Figure 1: Best performing models for feature-engineering approach. ALL, FCR, word bigrams, SE w/ SVM, and SE w/ LogReg when tested on the blind dataset. Character four-grams and five-grams when tested on the validation set. SE w/ SVM = Sentence-embeddings with SVM. SE w/ LogReg = Sentence-embeddings with Logistic Regression.

For BERT-like models, the highest accuracy of all six models is displayed in Figure 2. This included only models with all layers, as models with frozen layers demonstrated lower accuracy. The results show that two small BERT-like models and one large model performed with the highest accuracy: ALBERT (83%), DistilBERT-base-uncased (81%), and BERT-base-uncased (73%). As ALBERT and DistilBERT-base-uncased are lighter models, these results demonstrate that lighter models perform better than larger models for this studies data. Additionally, compared to previous BERT results, the BERT results in this study outperformed previously reported results 83% vs 80.8% (Lotfi et al., 2020), but lower than cross-corpora comparison accuracy of 85.3% when using SVM and FFNN base classifiers (Steinbakken and Gambäck, 2020).

For GPT-4, we performed 3 runs for with-label and no-label options with temperature set to 0.2. The accuracy when labels were provided was 92.9% for the first two runs – with only one file being mislabeled, and 85.7% for the third run. The average accuracy of the three runs was 90.48%. When no labels were given, GPT-4 attained an accuracy of 50% for all three runs. German was misclassified as Turkish and Russian, and Spanish was mislabeled as Italian, French, and Turkish.

Testing Set	N-gram Type	N-gram	Accuracy	F1-score
VS	Word	Bi	**0.93**	**0.93**
BDS	Word	Bi	**1.00**	**1.00**
VS	POS	Bi	0.87	0.86
BDS	POS	Bi	0.71	0.71
VS	Word	Tri	**0.93**	**0.93**
BDS	Word	Tri	0.86	0.85
VS	POS	Tri	0.80	0.80
BDS	POS	Tri	0.79	0.78
VS	Character	Four	**0.93**	**0.93**
BDS	Character	Four	0.79	0.78
VS	Character	Five	**0.93**	**0.93**
BDS	Character	Five	0.86	0.86

Table 4: Accuracy and F1 for n-gram models. The best models are in bold. VS = Validation Set. BDS = Blind Dataset.

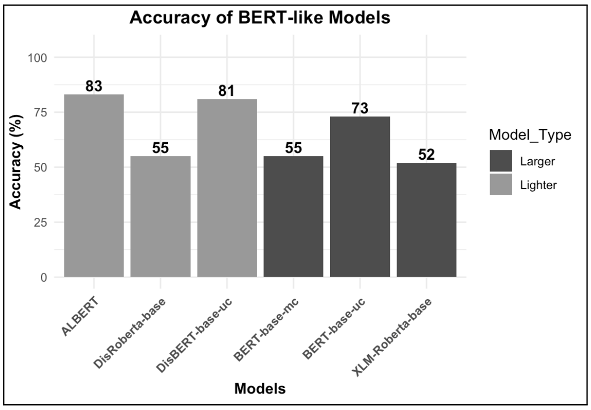


Figure 2: The results for BERT-like models. Dis = Distil. UC = Uncased. MC = Multilingual-cased.

5 Discussion

The contribution of the current study is two-fold. First, we compared two approaches - feature engineering and fine-tuning BERT-like LLMs - within the same study. The results showed that the feature-engineering approach outperformed the LLM-based approach, highlighting the effectiveness of feature-engineering pipelines for the NLI task, particularly in scenarios with relatively small datasets. Second, we explored a type of data that differs from that used in most previous studies. Specifically, our dataset consisted of narratives written by participants in response to the same video-based stimulus, providing more consistency across texts than the corpora of topic-based argumentative essays commonly used in NLI research.

Word bigrams were the most effective features extracted from the data. This finding suggested that word bigrams can effectively distinguish between learners with Spanish L1 and German L1 backgrounds based on their English writing. These n-grams likely captured differences in vocabulary use, word choices reflecting possible morphosyntactic errors, and distinctive lexical-syntactic patterns (the combinations of word tokens) between the two groups, which could be evidence of language transfer from learners' native languages to their L2 English. For example, German L1 influences were seen in lexical choices such as 'small human being' instead of 'baby' (possibly influenced by 'kleines menschliches Wesen' in German) and 'perceives it' instead of 'notices it' (possibly from 'wahrnehmen' meaning both 'perceive' and 'notice' in German).

Spanish L1 transfer was also evident from morphosyntactic patterns, such as noun-pronoun gender disagreement (e.g., 'the baby… she'). The preposition use was another source of transfer for Spanish L1 writers. For instance, 'yells him' (from Spanish 'le grita') reflected the incorrect omission of a preposition possibly due to the Spanish verb allowing a direct object.

An analysis of function words (Figure 3) revealed no major quantitative differences in the frequency of POS categories between the two groups, except for prepositions: German L1 writers tended to use more prepositions in their narratives compared to the Spanish L1 group. Qualitative differences in preposition use were seen, for instance, in 'walking on the street' phrase, where Spanish L1 writers overused the preposition 'on' instead of 'in'. The above examples indicated instances of linguistic transfer which are in line with the previous research on interlingual errors in Spanish-English bilinguals (Alonso Alonso, 1997). These patterns influenced the classifiers' decisions in disambiguating the two classes in the current study.

The Random Forest classifier also highlighted the bigrams that contributed to classification. For example, 'next to' was predominantly used by German L1 writers, while 'he is' and 'to leave' appeared more frequently in Spanish L1 texts. These features further illustrated the distinct lexico-syntactic choices between the two L1 groups. Overall, our results suggested that even when the differences between learner groups were subtle, traditional ML classifiers were capable of detecting them based on word n-grams and related surface-level patterns.

Importantly, our findings aligned with previous research that has identified word n-grams as effective features for NLI (e.g., Koppel et al., 2005; Jarvis et al., 2013) and demonstrated comparable or higher accuracy. For example, Jarvis et al. (2013) found that word and POS n-grams acquired an accuracy of 83.6% when using 10-fold cross validation and 90.1% when used on an ensemble classifier. However, our results cannot be directly compared because the number of classes and the nature of the data were different in the current study. In addition, word-based n-grams successfully captured class-specific differences from the dataset that consisted of written narratives based on the same video stimulus, thus reducing the risk of content bias from topic-related vocabulary.

Other n-gram types, including character n-grams and POS n-grams, also performed well. For character n-grams, we explored a range from four characters to nine characters. The best results were achieved with four- and five-grams. These likely captured class differences in short function words, such as prepositions, which are often markers of L1 influence (Jarvis and Odlin, 2000). The high performance of POS n-grams may be attributed to distinctive patterns in part-of-speech use and distribution across the two groups. For example, the qualitative analysis of the data suggested that German L1 writers relied more on subordinate clauses, a pattern consistent with transfer from German's preference for embedded structures (Swan and Smith, 2001).

Among lexical diversity and productivity features, the model combining all measures (function-to-content word ratio, MLU(w), TTR, and POS frequency counts) achieved the highest accuracy and

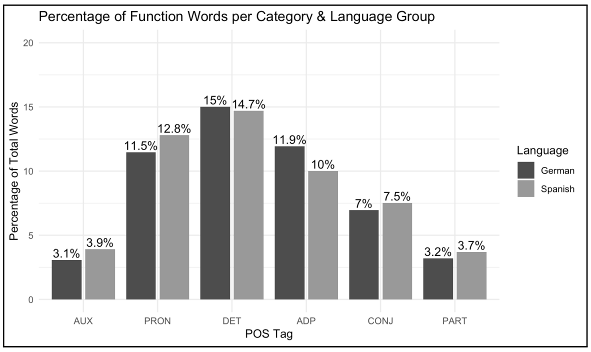


Figure 3: Percentage of function words per category and language group. AUX = Auxillaries. PRON = Pronouns. DET = Determiners. ADP = Adposition (Preposition). CONJ = Coordinating and Subordinating Conjunctions. PART = Particles.

F1 score (see Table 3). However, these results were still lower compared to the n-gram-based models. Notably, the function-to-content word ratio (FCR) emerged as the strongest individual predictor in this group, showing the highest performance on the blind test set. These patterns suggest that both n-gram features and FCR effectively captured differences in language productivity and distributional tendencies across German and Spanish L1 groups. Lexical diversity features, such as TTR, did not show high accuracy (50%) for the blind dataset. Exploring other TTR metrics (e.g., Moving-Average Type-Token Ratio (MATTR)) might provide a different result given the length-sensitive nature of the feature.

The sentence embeddings approach also outperformed the fine-tuning of BERT-like classification models with 93% accuracy. By encoding contextual relationships and sentence-level semantics, these embeddings were able to capture subtle differences in linguistic patterns between the two L1 groups in their English L2. These findings are in line with the previous research that indicated the utility of the embeddings approach for the NLI task and demonstrated that word embeddings together with string kernels were effective for L1 classification (Franco-Salvador et al., 2017).

Taken together, the results of the feature-engineering approach highlighted the robustness of both sparse vector surface-level features, such as n-grams, and dense sentence embeddings approach. Both methods were effective for distinguishing advanced learners' L1 backgrounds in written narratives.

The classification with BERT-like LLMs did not

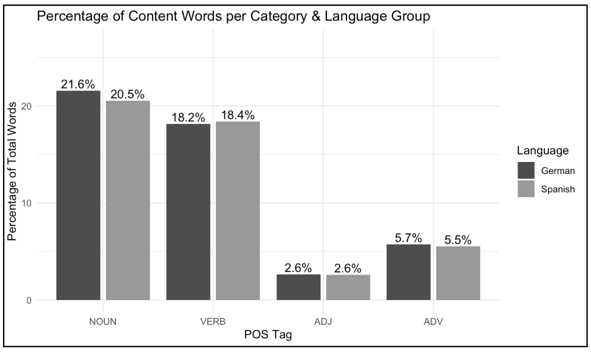


Figure 4: Percentage of content words per category and language group. ADJ = Adjective. ADV = Adverb.

perform on par with the feature-engineering approach. The highest accuracy within this group was achieved by the ALBERT model (83% accuracy, Figure 2), suggesting that lighter and more parameter-efficient architectures may be better suited for this task.

One possible explanation for the lower performance of BERT-like models is their sensitivity to dataset size and domain mismatch. Effective fine-tuning of these models typically requires large, diverse datasets to generalize better. In contrast, the relatively small and domain-specific nature of our dataset may have limited their ability to adapt. Additionally, while BERT models are designed for deep contextual understanding, this level of complexity may not be necessary for the current NLI task. Surface-level patterns, such as n-gram distributions and POS frequencies in our study, appear sufficient for distinguishing between L1 groups.

Furthermore, the results for the closed-source GPT-4 model revealed an average accuracy of 90.48%, which is similar to the sentence embeddings and word n-gram models. This performance was achieved using prompts that included labels, resembling a supervised approach. These findings align with previous studies investigating GPT models. For example, Zhang and Salle (2023) reported that GPT-4 achieved an accuracy of 91.7% on the TOEFL11 dataset. Similarly, Ng and Markov (2024) found that closed-source LLMs such as GPT-4 consistently outperformed open-source LLMs, regardless of fine-tuning. However, without labels, the closed-source GPT-4 performed poorly in our study.

Although both open- and closed-source models have demonstrated promising results for NLI, an important limitation of closed-source LLMs lies in the lack of transparency regarding their training data which raises concerns about reproducibility and potential biases in their outputs.

Overall, our results highlighted that traditional supervised machine learning techniques (e.g., SVM classifier) remain highly robust for low-resource NLI tasks. These models not only outperformed BERT-like LLMs but also achieved performance on par with the GPT-4 model. The lower results for BERT-like LLMs underscore their limitations in settings with domain-specific and scarce training data, including issues of limited interpretability and a higher risk of overfitting during fine-tuning.

6 Conclusion & Future Directions

In this paper, we compared two approaches for the NLI binary classification task: the traditional ML feature-engineering method and fine-tuning of BERT-like LLMs with a classification head. Our findings suggested that studies working with smaller, domain-specific datasets may benefit more from feature-engineering pipelines than from fine-tuning BERT-like LLMs. Frequency-based surface-level features were more sensitive to subtle differences in written narratives of similar content. While BERT-like models were less robust, lighter variants performed noticeably better than their larger counterparts on the small NLI dataset. Nonetheless, including other fine-tuning methods (e.g., DAPT, LoRA) could produce different results. The GPT-4 model also showed promising results when provided with labels; however, since the sources of its training data are not transparent, it is difficult to assess the generalizability and reliability of its performance. By evaluating both feature-engineering and BERT-like LLM approaches within the same study, we offered a direct comparison of their effectiveness for NLI.

Future studies could focus on datasets with structurally and topically consistent content across classes, which may reveal more subtle linguistic cues relevant for classification. It would also be valuable for future work to explore robust cross-validation techniques for LLMs, particularly when sufficient computational resources are available. Future research should continue to explore both traditional feature-engineering and LLM approaches, including closed-source LLM models without given labels, within the same experimental framework to better understand their comparative advantages across diverse domain-specific datasets.

References

Katsiaryna Aharodnik, Marco Chang, Anna Feldman, and Jirka Hana. 2013. Automatic identification of learners' language background based on their writing in czech. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1428–1436.

María Rosa Alonso Alonso. 1997. Language transfer: Interlingual errors in spanish students of english as a foreign language. *Revista alicantina de estudios ingleses, No. 10 (Nov. 1997); pp. 7-14*.

Stephen Pit Corder. 1975. Error analysis, interlanguage and second language acquisition. *Language teaching*, 8(4):201–218.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20:273–297.

Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232.

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.

Marc Franco-Salvador, Greg Kondrak, and Paolo Rosso. 2017. Bridging the native language and language variety identification tasks. *Procedia computer science*, 112:1554–1561.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*. John Wiley & Sons.

Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. Maximizing classification accuracy in native language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 111–118.

Scott Jarvis and Scott A Crossley. 2012. *Approaching language transfer through text classification: Explorations in the detectionbased approach*, volume 64. Multilingual Matters.

Scott Jarvis and Terence Odlin. 2000. Morphological type, spatial reference, and language transfer. *Studies in second language acquisition*, 22(4):535–556.

Ekaterina Kochmar. 2011. *Identification of a writer's native language by error analysis*. Ph.D. thesis, Master's thesis, University of Cambridge.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 624–628.

Robert Lado. 1957. *Linguistics across Cultures: Applied Linguistics for Language Teachers*. The University of Michigan Press.

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*.

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.

Cristóbal Lozano, Ana Díaz-Negrillo, and Marcus Callies. 2020. Designing and compiling a learner corpus of written and spoken narratives: Corefl. *What's in a Narrative*, pages 21–46.

Shervin Malmasi and Aoife Cahill. 2015. Measuring feature diversity in native language identification. In *Proceedings of the tenth workshop on innovative use of NLP for building educational applications*, pages 49–55.

Shervin Malmasi and Mark Dras. 2015. Large-scale native language identification with cross-corpus evaluation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1403–1409.

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.

Silvina Montrul. 2014. Interlanguage, transfer and fossilization: Beyond second language acquisition. In *Interlanguage*, pages 75–104. John Benjamins Publishing Company.

Yee Man Ng and Ilia Markov. 2024. Leveraging open-source large language models for native language identification. *arXiv preprint arXiv:2409.09659*.

Terence Odlin. 1989. *Language transfer*, volume 27. Cambridge University Press Cambridge.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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.

Michael Swan and Bernard Smith. 2001. *Learner English: A teacher's guide to interference and other problems*, volume 1. Cambridge University Press.

Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16.

Sze-Meng Jojo Wong and Mark Dras. 2009. Contrastive analysis and native language identification. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 53–61.

Wei Zhang and Alexandre Salle. 2023. Native language identification with large language models. *arXiv preprint arXiv:2312.07819*.

Systematic Evaluation of Rule-Based Analytics for LLM-Driven Graph Data Modelling

Fabio Yáñez-Romero
University Institute for Computer Research
University of Alicante
fabio.yanez@ua.es

Andres Montoyo
Armando Suárez
Department of Computing and Information Systems
University of Alicante
montoyo@dlsi.ua.es
armando@dlsi.ua.es

Alejandro Piad-Morffis
Yudivian Almeida-Cruz
School of Math and Computer Science
University of Havana
apiad@matcom.uh.cu
yudivian@matcom.uh.cu

Abstract

Artificial intelligence models have increasingly supplanted traditional rule-based systems for extracting knowledge from structured data; however, the integration of both approaches remains underexplored. While large language models offer greater flexibility than rigid rule systems, the structured knowledge from rule-based analytics can significantly enhance LLM performance and efficiency. This paper presents a novel multi-agent system that automatically generates graph database schemas from tabular data by strategically combining rule-based analytics with large language models. Our system utilises a lightweight rule framework that selects the most suitable analytical methods based on column data types, providing targeted insights to inform the schema generation process. The system's modular architecture enables comprehensive ablation studies examining both the effectiveness of rule-based analytics and their optimal presentation formats. Through systematic evaluation, we demonstrate that structured rule formats reduce result variability (lower standard deviation) while contextualised formats achieve superior performance despite higher variance. Our analysis identifies which pipeline stages benefit most from analytical guidance, providing insights for optimising hybrid AI systems. This work contributes a practical framework for integrating rule-based knowledge with modern language models, demonstrating measurable improvements in both consistency and performance for structured data processing tasks.

1 Introduction

The evolution of natural language processing has involved different rule-based (Miller et al., 1996), statistical (Weikum, 2002), and machine learning systems (Galanis et al., 2021), culminating in the current dominance of Large Language Models (LLMs) (Feng et al., 2025). However, recent approaches suggest that there is room for improvement with techniques traditionally used in rule-based systems when combined with LLMs (Laqrichi, 2024). While LLMs have revolutionised most NLP tasks with their exceptional reasoning capabilities, they still face challenges with complex linguistic phenomena, scalability, and domain-specific accuracy requirements (Gururaja et al., 2023). These limitations have revived interest in knowledge-based and rule-based approaches, which offer superior explainability and remain competitive in niche domains (Chen et al., 2025).

Rule-based analytics have been the cornerstone of classical information extraction from structured data (Atzmüller et al., 2008), involving the extraction of entities, properties, and relationships via discovered data types. However, these analytical methods, while interpretable and precise, lack the semantic interpretability necessary to accurately handle multi-column relationships and implied patterns.

Contemporary causal language models demonstrate a remarkable capacity to understand structured data formats such as CSV, JSON, and Markdown (Oh et al., 2025), enabling them to reason over tabular data when provided with appropriate context. For automatic generation of graph database schemas from relational ones, such a combination is particularly valuable. Relational databases represent entities as tables with primary keys and associated columns, and relationships as foreign keys. Although this structure guarantees coherence and integrity, it is not suitable for tasks involving the detection of implicit relationships, hierarchical understanding, or semantic flexibility—the essential ingredients for graph-based representations.

Our approach demonstrates how rule-based analytics can be integrated systematically with LLMs to address these challenges. We employ a rule-

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 154–164, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-016

based system that infers data types for every column and calls specialised analytical routines based on these type determinations. These analytics are then exposed as structured or contextualised context to LLMs in a multi-agent system, allowing us to contrast the relative performance impact of rule-based preprocessing on LLM-based schema generation.

The multi-agent system architecture enables systematic ablation studies by selectively masking analytical components, allowing us to quantify the contribution of rule-based analytics to overall system performance. Each agent specialises in different aspects: individual table analysis leveraging type-specific rules, cross-table relationship detection, and schema standardisation and integration.

2 Related Work

The automatic generation of graph database schemas from relational data represents a convergence of several fundamental research areas. Our work builds upon three interconnected domains: semantic interpretation techniques for extracting meaning from relational data, methodologies for converting relational schemas to graph representations, and the integration of large language models with tabular data processing.

2.1 Semantic Interpretation in Relational Data

The interpretation of semantics in tabular data has evolved significantly from early rule-based systems and heuristics (Cremaschi et al., 2024) to machine learning approaches (Chen et al., 2019). Traditional approaches relied primarily on unsupervised clustering techniques and supervised learning methods for column type classification and entity disambiguation. The introduction of dense vector representations marked a paradigm shift (Gorishniy et al., 2023), with specialised embedding techniques designed for tabular data enabling effective representation of column semantics, entity relationships, and cross-table linkages.

The emergence of large language models has fundamentally transformed semantic interpretation by enabling contextualised understanding of table content and structure (Cremaschi et al., 2025). Encoder-only models, such as BERT, have demonstrated effectiveness for header classification and column similarity assessment (Trabelsi et al., 2022). In contrast, decoder-only models such as Llama (Jiang et al., 2024) excel at entity linking, relationship extraction, and cross-table reasoning through in-context learning.

2.2 From Relational to Graph Databases

The conversion from relational to graph database schemas represents a critical challenge in modern data management (Bhandari and Chitrakar, 2024). While relational databases ensure data integrity through rigid schemas with primary keys, foreign keys, and predefined relationships, their structural constraints limit adaptability for downstream tasks requiring flexible semantic modelling.

Graph databases address these limitations by representing entities as nodes and relationships as edges, enabling more flexible modelling of semantic relationships. The conversion process involves identifying entities (potentially distributed across multiple tables), detecting implicit semantic relationships, and standardising properties and types. This transformation requires careful consideration of graph type selection (property graphs vs. RDF), structural properties (directionality, multigraphs), and higher-level semantic rules (Putrama and Martinek, 2022).

The complexity of this conversion process has motivated researchers to explore automated approaches leveraging advanced reasoning capabilities, leading to increased interest in utilising large language models for schema conversion (Sui et al., 2024a).

2.3 LLMs Integration with Tabular Data

Large Language Models have demonstrated remarkable capabilities in processing structured data through advanced prompt engineering techniques such as Chain-of-Thought reasoning (Wang et al., 2024) and in-context learning (Wen et al., 2025). However, several critical limitations constrain their effectiveness:

Format Sensitivity: LLMs exhibit pronounced sensitivity to tabular serialisation methods, with performance degradation of approximately 50% when tables are transposed (Liu et al., 2023). HTML and XML formats demonstrate superior performance with GPT models (Sui et al., 2024a).

Context Window Limitations: Context constraints pose significant challenges when processing larger tables, leading to performance degradation and the "lost-in-the-middle" phenomenon (Sui et al., 2024b).

Reliability Concerns: LLM outputs remain prone to hallucinations (Su et al., 2024), particularly in sensitive applications, with severity increasing as output length extends (Harrington et al., 2024). Mitigation strategies include audit modules and self-correction mechanisms (Karbasi et al., 2025).

External Tool Integration: The integration of external tools has significantly enhanced LLM utility for tabular data tasks, enabling code generation for database interaction (Zhang et al., 2023) and automated data processing workflows (Fan et al., 2024).

Despite these advances, current approaches primarily rely on commercial LLMs (Chen et al., 2025), limiting reproducibility and raising privacy concerns. Furthermore, existing methods lack a systematic evaluation of how rule-based analytics can enhance LLM performance in schema generation tasks, representing a significant gap that our work addresses.

3 MultiAgent System

To systematically evaluate the impact of rule-based analytics on graph schema generation from tabular data, we developed a multi-agent system that integrates data analytics with causal language models. Our primary objective is to generate valid graph database schemas from relational tabular data while enabling controlled experimentation to assess the contribution of rule-based preprocessing to overall system performance.

3.1 System Architecture and Design Principles

We implemented our system using LangGraph (Wang and Duan, 2024), a framework that enables the definition of distinct state graphs for different processing pipelines. This architectural choice provides crucial flexibility for our experimental design, allowing us to conduct ablation studies by selectively turning on or off specific nodes and analytics-driven prompts within the language model workflows. This modular approach facilitates systematic comparison between schema generation with and without rule-based analytical enhancement.

Our system architecture mirrors the decision-making process employed by expert graph database modellers when converting relational databases to graph representations (De Virgilio et al., 2013). The design incorporates domain expertise through a structured two-stage approach that addresses the inherent complexity of semantic interpretation and schema transformation. A comprehensive diagram illustrating the state graph used in our experiments, with and without analytics integration, is presented in Figure 1.

3.2 Processing Pipeline Architecture

The schema generation process operates through two complementary stages designed to capture both intra-table and cross-table semantic relationships:

1. **Table-Based Processing Pipeline**: This stage executes individual state graphs for each table in the source dataset, focusing on entity identification, relationship discovery, and property mapping within the context of each isolated table.

2. **Cross-Table Processing Pipeline**: This stage utilises a unified state graph to standardise redundant entities and relationships across tables, while identifying cross-table relationships, including primary and foreign key associations.

This dual-stage approach enables a systematic evaluation of how rule-based analytics influence various aspects of the schema generation process, ranging from local entity recognition to global schema coherence.

3.3 Table-Based Processing Pipeline

The table-level state graph implements three sequential processing nodes, each designed to leverage rule-based analytics for enhanced semantic understanding:

1. **Entity Identification**: Our system infers one or multiple entities within individual tables or recognises tables that lack sufficient information for entity extraction. When no entities are identified by the language model for a specific table, the table is excluded from the current pipeline stage.

2. **Intra-Table Relationship Discovery**: When multiple entities are detected within the same table, the language model infers relationships between those entities.

3. **Property Mapping**: For each column in a table, the system calls the language model to associate the column with identified entities or

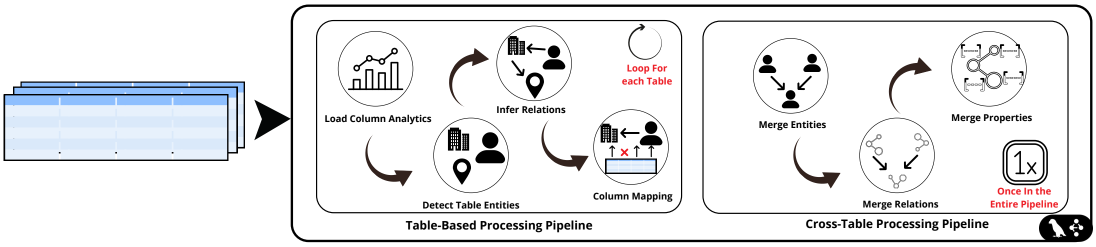


Figure 1: Entire Architecture for the system. Each table is processed individually before merging entities, relations and properties.

relationships. This process can be enhanced by providing the analytics related to that specific column.

The underlying strategy leverages the language model's ability to identify entities and relationships based on primary and foreign key analysis, enriched by rule-based analytics that provide deeper insights into column semantics and value distributions.

3.4 Cross-Table Processing Pipeline

The cross-table state graph operates on aggregated context from all processed tables to ensure schema consistency and completeness:

1. **Entity Standardisation**: The language model examines all previously identified entities, considering their names and associated properties through the initial columns and determine which semantically equivalent entities should be merged.

2. **Relationship Standardisation**: This process is activated when merged entities possess relationships with different names but equivalent semantic meanings. The model assigns the most appropriate name to these semantically equivalent relationships, ensuring schema coherence and reducing redundancy.

3. **Property Standardisation**: After entity and relationship standardisation, the system validates that all properties from merged components are correctly preserved and consolidated. The module identifies potential property conflicts arising from merging (such as duplicate properties with different data types) and applies resolution strategies to maintain schema integrity. This validation step is crucial for preserving the semantic richness captured during the table-based processing phase.

This systematic approach enables a precise evaluation of how rule-based analytics contribute to various aspects of schema generation, ranging from local semantic interpretation to global schema standardisation and consistency. The code for using the agent, as well as reproducing the entire experiment, can be found on GitHub[1].

4 Experimental Settings

Building upon the multi-agent system architecture described in the previous section, we designed a comprehensive experimental framework to systematically evaluate the impact of rule-based analytics on graph schema generation performance. Our experimental design enables controlled ablation studies that isolate the contribution of different analytical approaches to the overall effectiveness of the system.

4.1 Rule-Based Analytics Integration

The core hypothesis of our work centres on the premise that rule-based analytics can significantly enhance LLM performance in semantic interpretation tasks. To test this hypothesis, we implemented a type-specific analytical system that applies tailored analytics based on automatically inferred column data types. Our rule-based system categorises columns into four fundamental data types: categorical (including Boolean), string, numerical (including integer and float values) and date. The selection of specific analytics for each data type is grounded in established data science practices that optimise information extraction based on the inherent characteristics of each data type.

A detailed specification of the analytics performed for each data type is presented in Figure 2. These analytics range from basic statistical measures (mean, variance, distribution characteristics) to samples and automatically generated descrip-

[1] Repository for the Agentic Framework

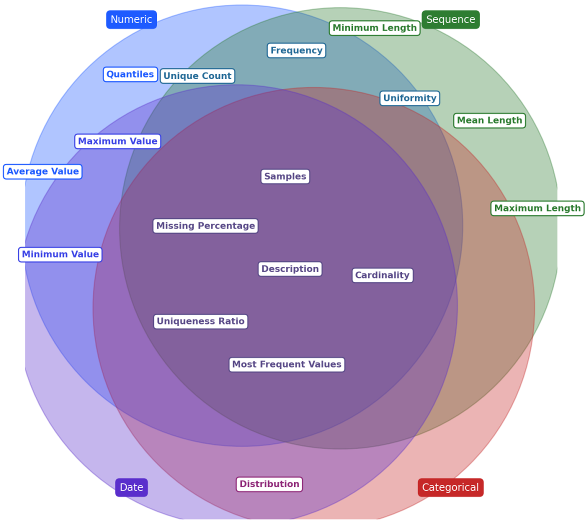


Figure 2: Analytics performed according to each data type detected. The intersections in the Venn Diagram represent the analytics that are shared among different data types.

tions of the entire columns, providing rich information for LLM decision-making.

4.2 Experimental Configurations

To systematically evaluate the contribution of rule-based analytics, we designed three distinct experimental configurations that represent different levels of analytical integration:

1. **Version 1 "No Analytics Baseline"** (V_1): This configuration serves as our baseline, providing only representative data examples for each column without any analytical context. This experiment enables direct measurement of the analytical contribution by comparing performance against pure LLM reasoning capabilities.

2. **Version 2 "Structured Analytic"** (V_2): This configuration provides comprehensive analytical results in a structured JSON format, exactly as computed and stored by our rule-based system. This approach tests the capability of the language model for understanding structural information while maintaining a clear organisational structure that facilitates systematic processing.

3. **Version 3 "Contextualised Analytic"** (V_3): This configuration applies analytical contextualisation methodologies inspired by successful approaches such as DeepJoin (Dong et al., 2023), which demonstrated significant improvements in semantic table interpretation through effective context integration. In this version, raw analytical results are transformed into natural language descriptions that provide semantic context about column characteristics, distributions, and relationships.

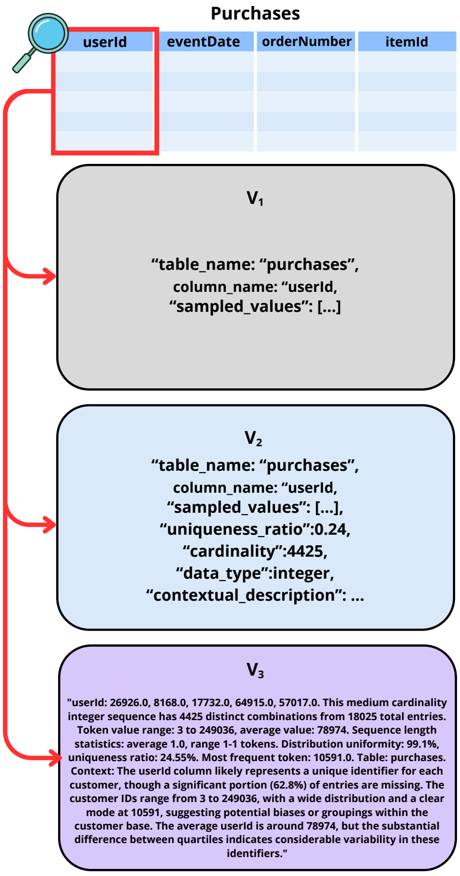


Figure 3: Analytics context formats supplied to the LLM-based agent for inferring the graph (property-graph) schema from the tabular dataset. V_2 encodes the analytics as structured JSON, while V_3 ("contextualised analytics") expresses the same information as narrative text generated deterministically from V_2 via a Python function to mimic typical LLM prompts. The figure contrasts these formats to assess how structured versus free-text context affects schema generation.

Importantly, all experimental variations utilise identical pipeline logic and differ only in the initial prompts provided to the language models. This design ensures that observed performance differences can be explicitly attributed to the presence or absence of rule-based analytics rather than architectural variations. A sample of each version is shown in Figure 3.

4.3 Implementation and Reproducibility Measures

To ensure experimental reproducibility and address the limitations of commercial LLM dependency identified in related work (Chen et al., 2025), we implemented our entire system using locally executed models. We selected Gemma 3 12B (Team et al., 2025) quantised to 4 bits, specifically the version hosted by Ollama[2], which provides an optimal balance between model capability and computational accessibility on standard user GPUs.

Our experimental configuration employs several measures to ensure reproducible results:

- Fixed Random Seed: All experiments use identical random seeds to ensure a consistent model behaviour across runs.
- Zero Temperature: Model temperature is set to zero to minimise stochastic variations in output generation.
- Model Consistency: The same model instance is used across all pipeline stages within each experimental run.
- Local Execution: All models are loaded and executed locally, eliminating external dependencies and ensuring data privacy.

Even with temperature = 0 and a fixed seed, LLM inference is not strictly deterministic: GPU-level numerical effects (e.g., parallel reductions, fused kernels, library autotuning) and decoder tie-breaking near probability ties can flip early tokens or the stopping point across runs (Atil et al., 2025), (Song et al., 2024). In our multi-agent pipeline, such micro-differences are amplified because each agent conditions on previous generations. We hypothesise that variability in answer length at early stages is the dominant driver: slightly longer/shorter completions change what downstream agents read, steering different trajectories and yielding different schema proposals. To address this, we ran 10 independent trials and report the mean and variance across runs.

4.4 Prompt Engineering Strategy

Our prompt design incorporates established techniques that have demonstrated effectiveness in structured data reasoning tasks. Specifically, we employ in-context learning examples that illustrate the desired schema generation behaviour, combined with Chain-of-Thought (CoT) reasoning prompts that guide the model through systematic analysis steps. This approach has proven particularly effective in interpreting tabular data, as demonstrated in recent literature (Liu et al., 2025). The complete prompt specifications for each experimental configuration are detailed in the experiment repository[3], enabling full reproducibility of our experimental setup. Each prompt variant maintains an identical logical structure while varying only in the analytical context provided to the language model.

4.5 Statistical Validation

To ensure the statistical significance of our results, each experimental configuration is executed ten times under identical conditions. We calculate both mean performance metrics and variance measures for each version of the experiment and for each dataset, enabling robust statistical analysis of the analytical contribution. This approach addresses the inherent variability in LLM outputs while providing sufficient statistical power to detect meaningful performance differences between analytical and baseline configurations. This experimental design directly addresses the research gap identified in our literature review regarding the systematic evaluation of rule-based analytics in LLM-driven schema generation tasks, providing a rigorous framework for assessing the effectiveness of our integrated approach.

4.6 Dataset

For our experimentation, we employ the Diginetica dataset, a large-scale benchmark released initially for the CIKM Cup 2016[4]. This dataset has become a cornerstone in session-based recommender system research due to its comprehensive coverage of real-world e-commerce interactions. Crucially for our purposes, the Diginetica dataset is organised into multiple interrelated tables, making it especially suitable for exploring the transition from a tabular to a graph-based data model:

- **Items:** Each product is uniquely identified and annotated with descriptive features such as price and textual tokens.

[2] Gemma 3 12B quantised model on Ollama

[3] Prompt Templates used in the experiments

[4] Original challenge where Diginetica Dataset was released

- **Categories:** Products are mapped to one or more categories, introducing a hierarchical structure that enriches the context for each item.
- **Views:** Every user interaction with a product page is captured, including session identifiers, temporal ordering, and user context.
- **Purchases:** Purchase events are linked to sessions and users, with references to related Items and Views, effectively connecting user actions across the dataset.
- **Queries:** This table logs user search activities with timestamps and contextual information, referencing entities from the other tables and enabling the reconstruction of full user search journeys.

The high degree of correlation and reference among these tables naturally aligns with the principles of graph data modelling, where entities (e.g., users, items, categories) become nodes and their relationships (such as views, purchases, and category memberships) are represented as edges. Such a structure facilitates the explicit modelling of complex interdependencies and interaction patterns that may be cumbersome to express or query efficiently in a purely tabular schema.

Therefore, Diginetica's rich, interconnected tabular design provides an ideal foundation for our task of translating traditional relational data into a graph database schema, enabling more expressive analysis and supporting advanced graph-based recommendation and user modelling techniques.

5 Results and Discussion

5.1 Evalutation Method

From the tabular dataset, we derived a lossless, agnostic property graph schema using Grok-4 (xAI, 2025). A graph data expert then reviewed and refined the naming, cardinalities, and data types to establish the expert-validated golden schema. We evaluated each experimental variant against this reference by measuring completeness (recall) over nodes, relationships, and properties; node/edge matching was synonym- and alias-aware to handle LLM naming variance, while property names were matched precisely to the original columns.

The completeness assessment methodology varied according to the schema component being evaluated:

- **Node completeness**: Measured by comparing the types of nodes present in the generated schema against those defined in the golden schema
- **Property completeness**: Assessed by determining whether nodes and relationships contain the properties they should possess based on the original relational database columns
- **Relationship completeness**: Evaluated based on whether relationships between existing node types match those in the golden schema, regardless of relationship names or directionality

The relationship evaluation methodology was deliberately simplified due to practical constraints. Language models frequently infer relationships with inverse orientations, incorrect directionality, or overly generic names. This complexity made the automatic evaluation of relationship completeness challenging and hindered the assessment of improvements in relationship detection across experimental versions.

5.2 Discussion of Results

The experimental results, presented in Table 1, show average outcomes and standard deviations across 10 independent tests per experimental version, along with the best-performing results for each version. Based on these findings, we can conclude the impact of column analytics usage and format on schema generation performance. The discussion is organised into specific component-level results and overall schema prediction performance.

5.2.1 Specific Results

Node Detection Performance: Node completeness showed minimal sensitivity to the use of analytics. When analytics were applied, unstructured sentence-format analytics proved counterproductive, with some contextualised analytics experiments degrading node type detection performance compared to baseline conditions.

Property Detection Performance: Property completeness, which depends solely on mapping columns to predefined entities, demonstrated a clear improvement with the use of analytics. Contextualised analytics format achieved the highest success rates in this component, suggesting that rich contextual information aids in accurate property-entity mapping.

Table 1: Completeness Percentage for Node, properties and relations, comparing the schema generated with the golden schema for Diginetica Dataset.

Completeness	**No Analytics** (V_1)	**Structured Analytics** (V_2)	**Contextualised Analytics** (V_3)
Node	$\mathbf{85.70 \pm 0.00}$	$\mathbf{85.70 \pm 0.00}$	82.84 ± 5.72
Property	70.87 ± 1.86	73.88 ± 3.74	$\mathbf{74.26 \pm 5.68}$
Relation	68.75 ± 13.98	63.75 ± 3.75	$\mathbf{75.00 \pm 11.18}$
Overall	75.11 ± 4.91	74.44 ± 1.11	$\mathbf{77.39 \pm 7.24}$

Relationship Detection Performance: Relationship completeness yielded mixed results across experimental conditions. Experiments without column analysis outperformed those using structured analytics, but underperformed compared to contextualised analytics approaches. This suggests a non-linear relationship between analytics complexity and the accuracy of relationship detection.

5.2.2 Overall Results

Overall, the best predictions were obtained using contextualised analysis (V_3), while the worst results were obtained using structured analytics. From the point of view of variability in results, the most uniform results are achieved between experiments using this set of structured analytics (V_2). In contrast, the most unpredictable results are obtained when the analytics are contextualised.

6 Discussion and Conclusion

The results indicate that while basic data analytics (providing representative column subsets along with column and table names) do not enhance node detection in inferred graph databases, they significantly improve property and relationship detection. Contextualised analytics demonstrated improvements of up to 9% in these components, with the format of contextual data proving critical for optimal relationship detection.

When evaluating overall schema generation effectiveness, contextualised analytics maximised model performance, while structured analytics yielded the poorest results. This suggests that rich, contextual information enables more accurate schema inference than rigid, structured data formats.

From a consistency perspective, structured analytics dramatically reduced result variability, as evidenced by lower standard deviations. This finding suggests that structured analytics should be preferred when result stability is prioritised over peak performance. Conversely, contextualised analytics produced the highest variability—exceeding even baseline conditions without analytics—making them the least stable approach across all experimental versions.

These findings present a clear trade-off between performance and stability in graph schema generation. Users prioritising maximum accuracy should employ contextualised analytics, despite increased result variability, while those requiring consistent, predictable outcomes may benefit from structured analytics approaches, albeit with reduced peak performance.

7 Limitations and Future Work

The experiments conducted present several limitations that we intend to address in future work, such as the use of open-source models of different sizes to verify the degradation/improvement based on model size.

Likewise, it would be of great interest to make a comparison with large commercial models, according to similar methodologies applied by previous works (Chen et al., 2025), which would give us an idea of what percentage of success can be expected with a multi-agent system like this compared to frontier models, being able to measure at this point also the computational cost associated with numerous calls of medium-sized models compared to the use of these commercial models.

On the other hand, the rule system used is extremely simple, with considerable room for improvement that can affect the final accuracy of the schema when determining the entities, relationships, and properties of the graph database.

Finally, truly understanding the limitations and capabilities of this system requires the use of more tabular data in various domains and with diverse characteristics, such as a large number of columns per table or tabular data that does not conform to the nomenclature of a relational database. In this sense, other structured formats provided for the analytics might be impactful on the final results, which needs further investigation.

Acknowledgments

This research has been funded by the University of Alicante, the Spanish Ministry of Science and Innovation, the Generalitat Valenciana, the Valencian Agency for Innovation (AVI), and the European Regional Development Fund (ERDF) through the following funding:"GeoIA: Artificial GeoIntelligence platform to solve citizens problems and facilitate strategic decision making in public administrations" (INNEST/2023/11), CORTEX (PID2021-123956OB-I00); funded by MCIN/AEI/10.13039/501100011033 and NL4DISMIS (CIPROM/2021/021).

References

Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2025. Non-determinism of "deterministic" llm settings.

Martin Atzmüller, Peter Klügl, and Frank Puppe. 2008. Rule-based information extraction for structured data acquisition using textmarker. In *LWA*.

Hira Lal Bhandari and Roshan Chitrakar. 2024. Enhancement of a transformation algorithm to migrate sql database into nosql graph database. *Data Science Journal*.

Jiaoyan Chen, Ernesto Jimenez-Ruiz, Ian Horrocks, and Charles Sutton. 2019. Learning semantic annotations for tabular data.

Zhikai Chen, Han Xie, Jian Zhang, Xiang song, Jiliang Tang, Huzefa Rangwala, and George Karypis. 2025. Autog: Towards automatic graph construction from tabular data.

Marco Cremaschi, Fabio D'Adda, and Andrea Maurino. 2025. steellm: An llm for generating semantic annotations of tabular data. *ACM Trans. Intell. Syst. Technol.* Just Accepted.

Marco Cremaschi, Blerina Spahiu, Matteo Palmonari, and Ernesto Jimenez-Ruiz. 2024. Survey on semantic interpretation of tabular data: Challenges and directions.

Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone. 2013. Converting relational to graph databases. In *First International Workshop on Graph Data Management Experiences and Systems*, GRADES '13, New York, NY, USA. Association for Computing Machinery.

Yuyang Dong, Chuan Xiao, Takuma Nozawa, Masafumi Enomoto, and Masafumi Oyamada. 2023. Deepjoin: Joinable table discovery with pre-trained language models.

Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Workflowllm: Enhancing workflow orchestration capability of large language models.

Chen Feng, Yifan Li, Zhaoda Chen, and Longxing Guo. 2025. The evolution and breakthrough of natural language processing: The revolution from rules to deep learning. In *Proceedings of the 2024 5th International Conference on Computer Science and Management Technology*, ICCSMT '24, page 307–311, New York, NY, USA. Association for Computing Machinery.

N. I. Galanis, P. Vafiadis, K. G. Mirzaev, and G. A. Papakostas. 2021. Machine learning meets natural language processing – the story so far.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. 2023. On embeddings for numerical features in tabular deep learning.

Sireesh Gururaja, Amanda Bertsch, Clara Na, David Widder, and Emma Strubell. 2023. To build our future, we must know our past: Contextualizing paradigm shifts in natural language processing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 13310–13325. Association for Computational Linguistics.

Fiona Harrington, Elliot Rosenthal, and Miles Swinburne. 2024. Mitigating hallucinations in large language models with sliding generation and self-checks.

Zhengyong Jiang, Jionglong Su, Tong Chen, Zimu Wang, and Procheta Sen. 2024. Knowledge base-enhanced multilingual relation extraction with large language models. In *LKM2024: The First International OpenKG Workshop Large Knowledge-Enhanced Models @IJCAI 2024*.

Amin Karbasi, Omar Montasser, John Sous, and Grigoris Velegkas. 2025. (im)possibility of automated hallucination detection in large language models.

Safae Laqrichi. 2024. A hybrid framework for cosmic measurement: Combining large language models with a rule-based system.

Si-Yang Liu, Qile Zhou, and Han-Jia Ye. 2025. Make still further progress: Chain of thoughts for tabular data leaderboard.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023. Rethinking tabular data understanding with large language models.

Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. 1996. A fully statistical approach to natural language interfaces. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 55–61, Santa Cruz, California, USA. Association for Computational Linguistics.

Jio Oh, Geon Heo, Seungjun Oh, Hyunjin Kim, JinYeong Bak, Jindong Wang, Xing Xie, and Steven Euijong Whang. 2025. Better think with tables: Tabular structures enhance llm comprehension for data-analytics requests.

I Made Putrama and Péter Martinek. 2022. An automated graph construction approach from relational databases to neo4j. In *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)*, pages 000131–000136.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024a. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. 2024b. Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D. Davison, and Jeff Heflin. 2022. Strubert: Structure-aware bert for table search and matching. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 442–451, New York, NY, USA. Association for Computing Machinery.

Jialin Wang and Zhihua Duan. 2024. Agent ai with langgraph: A modular framework for enhancing machine translation using large language models.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding.

Gerhard Weikum. 2002. Foundations of statistical natural language processing. *SIGMOD Rec.*, 31(3):37–38.

Xumeng Wen, Shun Zheng, Zhen Xu, Yiming Sun, and Jiang Bian. 2025. Scalable in-context learning on tabular data via retrieval-augmented large language models.

xAI. 2025. Grok 4. https://docs.x.ai/docs/models/grok-4-0709. Large language model by xAI.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M. Patel. 2023. Reactable: Enhancing react for table question answering.

Improved Contrastive Learning over Commonsense Knowledge Graphs for Unsupervised Reasoning

Rongwen Zhao and **Jeffrey Flanigan**
University of California, Santa Cruz
{rzhao17,jmflanig}@ucsc.edu

Abstract

Knowledge-augmented methods leverage external resources such as commonsense knowledge graphs (CSKGs) to improve downstream reasoning tasks. Recent work has explored contrastive learning over relation-aware sequence pairs derived from CSKG triples to inject commonsense knowledge into pre-trained language models (PLMs). However, existing approaches suffer from two key limitations: they rely solely on randomly sampled in-batch negatives, overlooking more informative hard negatives, and they ignore additional plausible positives that could strengthen training. Both factors limit the effectiveness of contrastive knowledge learning. In this paper, we propose an enhanced contrastive learning framework for CSKGs that integrates **hard negative sampling** and **positive set expansion**. Hard negatives are dynamically selected based on semantic similarity to ensure the model learns from challenging distinctions, while positive set expansion exploits the property that similar head entities often share overlapping tail entities, allowing the recovery of missing positives. We evaluate our method on unsupervised commonsense question answering and inductive CSKG completion using ConceptNet and ATOMIC. Experimental results demonstrate consistent improvements over strong baselines, confirming that our approach yields richer commonsense-aware representations and more effective knowledge injection into PLMs.

1 Introduction

Commonsense reasoning is fundamental for enabling machines to form assumptions about everyday situations and draw conclusions aligned with human understanding of commonly known facts (Davis and Marcus, 2015; Sap et al., 2020). Despite significant progress in natural language processing (NLP), endowing models with robust commonsense reasoning abilities remains an open challenge. This challenge has received growing attention in recent years with the release of versatile benchmark datasets targeting different aspects of commonsense reasoning. For example, Figure 1 illustrates a sample from the SocialIQA dataset (Sap et al., 2019b), which focuses on reasoning about human actions and their social implications. In parallel, the development of large-scale commonsense knowledge graphs (CSKGs), such as ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019a), has motivated tasks like inductive CSKG completion to further test models' ability to generalize over unseen entities (Malaviya et al., 2020; Wang et al., 2021).

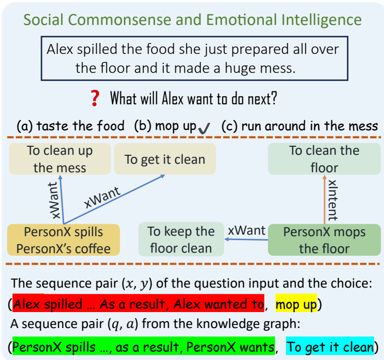


Figure 1: An example from a SocialIQA task focusing on reasoning about actions and social implications (**top**) (Sap et al., 2019b), with the relevant social commonsense knowledge triplets from ATOMIC (**middle**) (Sap et al., 2019a). The **bottom** shows a (input, choice) sequence pair of the example and a (premise, alternative) sequence pair of a knowledge graph triplet.

With the advent of large pre-trained language models (PLMs) (Devlin et al., 2018; Liu et al., 2019; Radford et al., 2019), fine-tuning PLMs on task-specific commonsense question answering (CSQA) datasets has led to strong results, in some cases approaching or surpassing human performance (He et al., 2020). However, reliance on large-scale human-annotated training data poses challenges, as such annotations are expensive and difficult to scale (Shwartz et al., 2020; Banerjee and Baral, 2020; Bosselut et al., 2021; Sun et al., 2022). Moreover, evidence shows that PLMs often

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 165–178, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-017

exploit spurious correlations or shortcuts in data (Branco et al., 2021), rather than performing genuine commonsense reasoning or effectively leveraging external knowledge sources (Banerjee et al., 2021).

To mitigate these limitations, several unsupervised approaches based on CSKGs have been proposed. For instance, Ma et al. (2021); Kim et al. (2022) generate synthetic QA pairs from CSKG triples by treating the head entity with its relation as a query and the tail entity as the gold answer. Yet, the coverage of such methods is constrained by the incompleteness of CSKGs (Ju et al., 2022). More recently, Su et al. (2022) introduced a contrastive learning framework that pre-trains PLMs on (premise, alternative) pairs synthesized from CSKGs. While effective, this approach has two major shortcomings: (i) it relies on randomly sampled in-batch negatives, overlooking the importance of *hard negatives*, and (ii) it ignores potentially valuable positive examples inherent in CSKG structures. Both factors may limit the efficacy of the contrastive learning paradigm.

In this work, we propose an enhanced contrastive learning framework to better utilize CSKGs for commonsense knowledge representation. Our method incorporates two key components: **(i) hard negative sampling**, which dynamically selects informative negatives that are neither trivial nor indistinguishably similar, and **(ii) positive set expansion**, which leverages the property that similar head entities in CSKGs often share overlapping tail entities, thereby recovering missing positives. By integrating these mechanisms into the contrastive objective, we more effectively exploit the structure of CSKGs to improve knowledge injection into PLMs.

We evaluate our framework on two widely used CSKGs, ConceptNet and ATOMIC, across unsupervised CSQA benchmarks, including COPA (Roemmele et al., 2011), SIQA (Sap et al., 2019b) and CSQA (Talmor et al., 2019) and inductive CSKG completion tasks. Experimental results demonstrate consistent improvements over strong baselines, confirming that our framework generates superior commonsense-aware knowledge representations.

2 Preliminaries and Preprocessing

In this section, we first introduce some preliminaries used in this work. Then we will present the preprocessing details.

2.1 Task Definition

Our task is the following: given a common-sense knowledge graph $\mathcal{G}$ and a pre-trained language model $\mathcal{M}$, we construct a synthesized corpus of sequence pairs $\mathcal{D} = \{(p_1, a_1), ..., (p_i, a_i)\}$ from $\mathcal{G}$, where p is the head sequence and a is the natural language description of the tail entity. Then we further train $\mathcal{M}$ on the corpus $\mathcal{D}$ so $\mathcal{M}$ performs better on a given downstream commonsense-related task represented as $\mathcal{T} = \{(x_1, y_1), ..., (x_m, y_m)\}$ by encouraging M to generate superior commonsense-aware knowledge representation embeddings for the sequence pair (x_m, y_m). The corpus $\mathcal{D}$ is constructed from $\mathcal{G}$ using the method described in §2.3.

2.2 Notation

We define our commonsense knowledege graph $\mathcal{G}$ as a 4-tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{P})$, where the vertices are entities $\mathcal{E}$ and $\mathcal{R}$ are the set of relation types. $\mathcal{T}$ is the set of all edges, where each edge is a triple (h, r, t). $h \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{E}$ are the head entity, relation, and tail entity, respectively. $\mathcal{P}$ is the collection of all relations expressed in natural language, as shown in Appendix A.2. Additionally, following previous work (Ouyang et al., 2021; Su et al., 2022) we augment $\mathcal{G}$ with inverse edges: for each edge triple $(h, r, t) \in \mathcal{T}$ we add its reverse triple (h, r^{-1}, t) into $\mathcal{G}$.

2.3 Knowledge Graph Triple to Natural Language

In CSKGs, the entities h and t in $\mathcal{E}$ are in a free-form text format, and the relation r is a specific word or short phrase based on the corresponding CSKG. For example, (h, r, t) in ConceptNet could be (*Bottle, MadeOf, Plastic*) or (*PersonX spills PersonX's coffee, xWant, To get it clean*) in ATOMIC. . We use a set of templates for the relation r and its reverse relation r^{-1} in ATOMIC and ConceptNet. Following previous work (Hwang et al., 2021; Huang et al., 2021; Su et al., 2022), we first convert each edge triple (h, r, t) into a sequence pair (p, a) in natural language, consisting of a head sequence and its tail sequence. The original relation r is converted to the pre-defined natural language template and then connect it with the head entity h to form the head sequence p, while a is the natural language description of the tail entity t.

For example, in Figure 1, for the head node "*PersonX spills PersonX's coffee*", we concatenate it

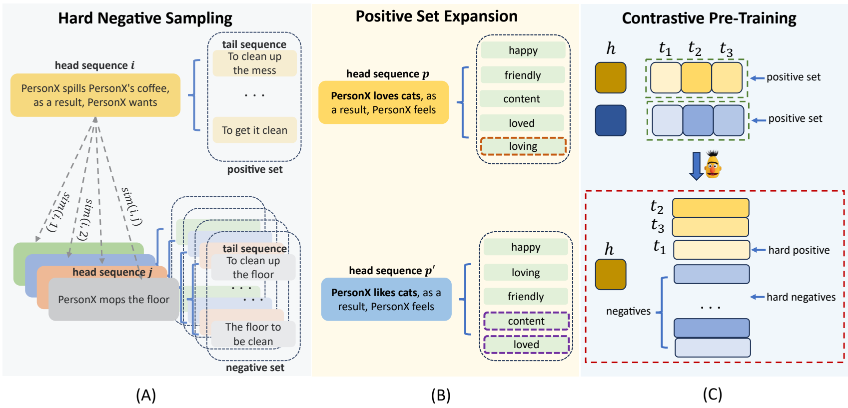


Figure 2: The steps in our contrastive learning framework. (A) **Hard Negative Sampling**: We dynamically sample hard negatives by the similarity of premise pairs. (B) **Positive Set Expansion**: We deliberately utilize the characteristic within the CSKGs that similar head entities are likely to share the same set of positive tail entities and expand the possible positive set mutually. (C) **Contrastive Training**: We integrate the updated sequence pairs into the existing multi-view contrastive learning framework to perform knowledge injection.

with the relation template of "xWant", resulting in the head sequence "*PersonX spills PersonX's coffee, as a result, PersonX wants.*" Similarly, for the reverse relation r^{-1}, we can also derive a sequence pair. Since for a head entity h, given a relation r, it may have n tail entities $\{t_1, t_2, ..., t_n\}$. Therefore, for a head sequence p, it may have a set of tail sequences $\{a_1, a_2, ..., a_n\}$.

2.4 Embedding Representation

After obtaining the sequence pair (p, a), we use a pre-trained language model (PLM) to get an initial embedding representation for the sequence pair. Specifically, for a sequence pair (p, a), where both p and a consist of sequence of tokens $\{x_0, ..., x_m\}$ and $\{y_0, ..., y_n\}$, respectively, We apply a PLM encoder to obtain the last hidden states of p and a, then use the hidden state of the first token, e_p and e_a as the embedding representation for p and a.

For a positive sequence pair (p, a), their representations in embedding space e_p and e_a should be close. We adopt the cosine similarity function to measure the distance of p and a:

$$sim(p, a) = cos(e_p, e_a)$$

3 Methodology

Our commonsense-aware knowledge representation learning framework, as shown in Figure 2, is divided into three steps: hard negative set sampling, positive set expansion, and contrastive knowledge fine-tuning. The input consists of a CSKG (e.g., ATOMIC) and a PLM (e.g., RoBERTa-Large). Given the synthesized CSKG sequence pairs obtained from §2.4, the goal is to inject the commonsense knowledge into the PLM by further training on the synthesized sequence pairs with enhanced contrastive learning.

We propose to enhance the existing contrastive learning framework for learning commonsense knowledge representation (Su et al., 2022). We propose two mechanisms to mitigate two issues that may impede the learning efficacy of the contrastive learning framework. First, we propose hard negative sampling to pay more attention to the hard ones instead of merely relying on random in-batch negatives (§3.1). Second, we propose to expand the positive set so that the missing positives could be recovered (§3.2). Finally, the PLM is trained with the adapted contrastive objective (§3.3).

3.1 Hard Negative Sampling

In this paper, we propose adapting the idea of hard negative sampling to the existing contrastive learning framework for the common sense-aware knowledge representation task. The learning framework learns commonsense knowledge representation with the contrastive information of the natural language sequence pairs. In particular, the existing method utilizes samples within the same mini-batch as negatives (Su et al., 2022), although such a strategy can significantly enhance training efficiency by repeatedly using the representations of in-batch negatives. However, this method ignores the difference of easy and hard negatives. Some literatures have theoretically and empirically proved that easy samples contribute less to the final learned representation (Bucher et al., 2016; Wu et al., 2017; Robinson et al., 2020; Zhang and Stratos, 2021). Recently, several adaptations in knowledge graph representation learning for knowledge graph completion and commonsense question answering also verify the importance of sampling hard negatives (Wang et al., 2022; Peng et al., 2022; Zhang and Li, 2022). The success of the contrastive representation benefits more from the hard ones, which means that the negatives that are difficult to distinguish are preferred instead of relying on randomly selected in-batch negatives.

To illustrate the proposed idea more precisely, consider the corpus $\mathcal{D}$ consisting of all triples converted from the CSKG $\mathcal{G}$ by the aforementioned steps and a given (p, a) from $\mathcal{D}$. The goal is to find hard samples (p', a') so that the model has difficulty differentiating the pair (p, a') in the latent embedding space. We propose to select hard negatives by the similarity between p and p' to form a hard negative set. For a sample (p') from $\mathcal{D}$, we first calculate the similarity $sim(p, p')$ between p and p'. If $\alpha < sim(p, p') < \beta$, where α and β are hyperparameters, then p' will be added into the set $\mathcal{I}^-$. We don't want to select negative examples too close to the positive example, so we have $sim(p, p') < \beta$, and we don't want examples that are too easy, so we have $\alpha < sim(p, p')$. Based on manual observations, we set $\alpha = 0.3$ and $\beta = 0.7$. We use the cosine similarity function to measure the similarity of p and p'.

An illustration of how we construct the negative samples is shown on the left in Figure 2. Let $A(p)$ be the collection of all tail entities $\{a_j, a_j, ..., a_j\}$ from $\mathcal{D}$ such that each tail sequence has the same head p. For each $p_j \in \mathcal{I}^-$ we obtain the head sequence and tail sequence pairs $(p_j, a_{j,o})$, where $a_{j,o} \in A(p_j)$ is the collection of all tail entities $\{a_{j,1}, a_{j,2}, ..., a_{j,n}\}$ from $\mathcal{D}$ such that each tail sequence has the same head p_j. The union of these sets forms our hard negative set.

3.2 Positive Set Expansion

We propose to expand the positive set by utilizing the unique property of CSKGs to incorporate some potential while valuable positives.

Specifically, given a sequence pair of head and tail set (p, a_i), $a_i \in A(p)$, we measure the similarities of p with other head sequences p'. The p' with the highest similarity $sim(p, p')$ will be selected. Then, given the similar head sequence p', $A(p)$ and $A(p')$ may share some tail sequences. For example, in Figure 2, for the head sequences "PersonX loves cats, as a result, PersonX feels" and "PersonX likes cats, as a result, PersonX feels", both have same tail sequences while contain their own exclusive ones. Hence, we propose heuristically expanding the positive set A by inserting the missing tail sequences obtained from the tail sequence set $A(p')$.

3.3 Training Objective

For the sample (p_i, a_i), we use the InfoNCE loss with additive margin (Chen et al., 2020; Gao et al., 2021):

$$L_i = -\log \frac{e^{(\phi(\mathbf{p_i}, \mathbf{a_h}) - \gamma)/\tau}}{e^{(\phi(\mathbf{p_i}, \mathbf{a_h}) - \gamma)/\tau} + \sum_{\mathbf{j=1}}^{|\mathcal{I}^-|} \sum_{o=1}^{k} e^{\phi(\mathbf{p_i}, \mathbf{a_{j,o}})/\tau}},$$

where the scoring function for a candidate sequence pair $\phi(\mathbf{p_i}, \mathbf{a_h}) = sim(\mathbf{p_i}, \mathbf{a_h})$. We use cosine similarity for our similarity function. For the hard positive, we select the one positive alternative a_h from the expanded set A which has the lowest similarity to p. The positive additive margin γ incentivizes the model to boost the score of the positive sequence pairs. By adjusting the temperature τ, the relative significance of negatives can be modified. A smaller value of τ increases the emphasis on challenging negatives, yet it also poses a risk of over-fitting to label noise.

3.4 Fine-Tuning Details

In practice, we fine-tune RoBERTa-Large (Liu et al., 2019) on the synthesized CSKG sequence pairs. The contrastive fine-tuning process directly equips the PLM with relation-aware commonsense knowledge, which can then be evaluated in zero-shot settings for commonsense QA and CSKG completion.

4 Experiments

In this section, we first introduce the CSKGs that we used in this study. Then we will present three evaluation tasks, unsupervised CSQA, inductive CSKG completion and claim verification, by intruding related benchmark datasets, baselines and main results. We conduct all experiments in a zero-shot setting, which means we do not have access to the official training data.

4.1 Commonsense Knowledge Graphs

Our experiments rely on two representative CSKGs, ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019a). Each KG has different knowledge types. Following previous work(Wang et al., 2021; Su et al., 2022), we use CN-82K and ATOMIC in our experiments. The statistics are shown in Table 7. Details of the CSKGs are listed in Appendix A.1.

4.2 Unsupervised CSQA

In this section, we evaluate our framework on commonsense question answering datasets in an unsupervised way, which can be formalized as follows: given a question q and a set of answer candidates A, the model could choose the most likely candidate $\hat{a}$ by $\hat{a} = \arg\max_{a \in A} \mathrm{sim}(\mathbf{q}, \mathbf{a})$, where $\mathbf{q}$ and $\mathbf{a}$ are representations obtained from the model.

Benchmarks: We conduct experiments on three different commonsense question answering datasets , COPA (Roemmele et al., 2011), SIQA (Sap et al., 2019b) and CSQA (Talmor et al., 2019) to verify the effectiveness of the proposed framework. Details of the datasets are listed in Appendix A.3.

Baselines: We compare the proposed framework with four different groups of baselines: (1) Vanilla PLMs (RoBERTa-Large (Liu et al., 2019), GPT2-L/M (Radford et al., 2019)); (2) Methods without relying on external CSKGs, instead by using PLMs to generate intermediate outputs (SEQA (Niu et al., 2021), self-talk (Shwartz et al., 2020), Dou (Dou and Peng, 2022)); (3) Prompting the large LMs to generate relevant knowledge given few-shot human annotations, including GKP (Liu et al., 2022) and TSGP (Sun et al., 2022); and (4) Models using CSKGs, including KTL (Banerjee and Baral, 2020), DynaGen (Bosselut et al., 2021), NLI-LM (Huang et al., 2021) and MICO (Su et al., 2022), a multi-view contrastive learning based baseline. For the details of each baseline method, please refer to their original papers. We are aware that there exist some other methods or method variants achieving better performance compared to the baselines listed here. However, they are either using larger backbone models (Sun et al., 2022) or trained with the larger even multiple knowledge bases (Ma et al., 2021; Kim et al., 2022). Both factors can improve the performance. Thus, we compare to methods with a similar model size as ours and the same knowledge bases. We also consider the issue of model size in §5.

Main Results: Table 1 shows the zero-shot evaluation results on benchmark datasets. Our model achieves the best performance across all baseline models on all datasets.

First, we compare our model with the vanilla PLMs, RoBERTa-Large (Liu et al., 2019), GPT2-L/M (Radford et al., 2019). It is not surprising that the LMs show significant and systematic performance gains on all datasets compared to the random baselines. Since it has been verified that the LMs already store implicitly vast amount of various types of knowledge in their parameters, such as relational and commonsense knowledge, which are universally indispensable for downstream tasks (Petroni et al., 2019).

Second, we compare our model with the methods generating intermediate outputs in the inference stage, such as SEQA (Niu et al., 2021) and self-talk (Shwartz et al., 2020). SEQA first generates a set of plausible answers and then compute the semantic similarity between each plausible answer and answer candidate. While self-talk iteratively queries the LMs with a set of information-seeking questions to disclose the potential background knowledge. However, this kind of methods cannot maintain their effectiveness systematically, even their performance is lower than the LM baselines. For example, as shown in Table 1, on CSQA dataset, self-talk is 8% lower than GPT2-Large, suggesting that self-talk may generate some spurious or misleading background knowledge. This shows that the explicit commonsense knowledge may be necessary to mitigate the hallucinations of LMs' generated knowledge. In light of this, our model injects explicit commonsense knowledge by self-supervising LMs on CSKGs. As shown in the results, our model can generate better commonsense knowledge representation advancing the unsupervised CSQA tasks.

Our method can achieve consistent improvement just by using relatively small backbone model. Compared with methods suzch as GKP (Liu et al., 2022) and TSGP (Sun et al., 2022), our best model outperforms them on SIQA and CSQA tasks without relying on large language models (LLMs). Similar as chain-of-thought (Wei et al., 2022), both GKP and TSGP first prompt the LLMs (GPT-3 and GPT2-XL, respectively) with few-shot human annotations to generate relevant background knowledge. However, knowledge snippets in nat-

Methods	Models	Knowledge Source	COPA dev	COPA test	SIQA dev	CSQA dev
Random	-	-	50.0	50.0	33.3	25.0
RoBERTa-L	RoBERTa-L	-	54.8	58.4	39.8	31.3
GPT2-L	GPT2-L	-	62.4	63.6	42.8	40.4
SEQA	GPT2-L	GPT2-L	-	-	46.6	34.6
self-talk	GPT2-[Distil/XL/L]	GPT2-[Distil/L/M]	66.0	-	46.2	32.4
Dou	ALBERT-XXL-v2	ALBERT-XXL-v2	-	-	44.1	50.9
GKP	T5-11b	few-shot exemplars + GPT-3	-	-	-	47.3
TSGP	GPT2-XL	few-shot exemplars + GPT2-XL	-	-	51.5	49.1
KTL	RoBERTa-L	ATOMIC	-	-	46.6	36.8
DynaGen	GPT2-M	COMET	-	-	50.1	-
NLI-LM	RoBERTa-L	ATOMIC+QNLI	-	-	-	52.1
MICO-CN	RoBERTa-L	ConceptNet	73.2	75.2	44.6	51.0
MICO-ATOMIC	RoBERTa-L	ATOMIC	79.4	77.4	56.0	44.2
Ours	RoBERTa-L	ConceptNet	73.8	77.2	46.2	**53.2**
Ours	RoBERTa-L	ATOMIC	**82.0**	**79.4**	**56.7**	47.8

Table 1: Accuracy (%) of unsupervised CSQA task on three public benchmarks. Our best scores are highlighted in bold.

ural language may not be sufficient to answer a commonsense-related question, since even LLMs still suffer from hallucination (Wei et al., 2022).

Our method can fine-tune LMs on CSKGs in a more effective and efficient way. Compared with methods using external CSKGs, such as KTL (Banerjee and Baral, 2020), DynaGen (Bosselut et al., 2021), NLI-LM (Huang et al., 2021) and MICO (Su et al., 2022), our method can achieve better performance even trained with the same CSKG. For a knowledge triplet, given knowledge representations of any two, KTL learns to generate the third one. While our method focuses on generating relation-aware contextualized representation given two sequence pairs. DynaGen dynamically generates contextually-relevant commonsense knowledge graphs by using a generative neural commonsense knowledge model, COMET (Bosselut et al., 2019). While the generated commonsense inferences are more context-relevant, it requires iterative generation that may impact the inference efficiency. Our method is more efficient by just generating contextually-relevant commonsense representations and selecting the most probable based on the largest similarity. NLI-LM utilizes extra NLI resources while unnecessary for our method. Our method outperform NLI-LM slightly by 1.1% on CSQA dataset. MICO is the most relevant to our method. It also utilizes contrastive multi-view training on CSKGs, while our method can bring consitent performance gains on all datasets compared with it. It shows the effectiveness of the two proposed modules, positive set expansion and hard negative sampling.

Model	ConceptNet MRR	ConceptNet Hits@10	ATOMIC MRR	ATOMIC Hits@10
ConvE	0.21	0.40	0.08	0.09
RotatE	0.32	0.50	0.10	0.12
Malaviya	12.29	19.36	0.02	0.07
InductivE	**18.15**	**29.37**	2.51	5.45
MICO	10.92	22.07	8.13	15.69
Ours	9.65	19.97	**8.29**	**15.93**

Table 2: Results on inductive CSKG completion. The best scores are highlighted in bold.

KG	Method	COPA dev	COPA test	SIQA dev	CSQA dev
Concept Net	Ours	73.8	77.2	**46.2**	**53.2**
	-w/o HNS	72.2	76.8	43.6	52.0
	-w/o PSE	**74.0**	**77.4**	43.9	52.7
ATOMIC	Ours	**82.0**	79.4	**56.7**	**47.8**
	-w/o HNS	79.0	**80.4**	56.0	44.4
	-w/o PSE	80.4	78.4	56.5	45.9

Table 3: Ablation study. The best scores are highlighted in bold.

4.3 Inductive CSKG Completion

Knowledge graphs, especially CSKGs, are often incomplete with missing entities and relations. Inductive CSKG completion evaluates the inductive capability of a model to predict relations triples for new, unseen entities (Wang et al., 2021). Given a knowledge triplet (h, r, t), the model needs to predict the unseen tail entity t by $(h, r, ?)$ or the unseen head entity by $(?, r^{-1}, t)$. Same as the previous work (Wang et al., 2021), we adopt the link predic-

Backbone	KG	COPA dev	COPA test	SIQA dev	CSQA dev
BERT Base	-	45.4	46.4	37.1	21.5
	ConceptNet	63.8	66.4	38.9	**43.2**
	ATOMIC	**69.8**	**74.0**	**48.2**	42.7
BERT Large	-	47.4	46.8	37.2	20.4
	ConceptNet	64.4	73.2	41.7	**47.8**
	ATOMIC	**73.2**	**74.2**	**51.6**	43.9
RoBERTa Base	-	52.0	55.2	38.4	29.2
	ConceptNet	62.4	69.6	40.1	**45.4**
	ATOMIC	**72.4**	**73.4**	**52.1**	41.0
RoBERTa Large	-	55.0	58.6	39.8	31.3
	ConceptNet	73.8	77.2	46.2	**53.2**
	ATOMIC	**82.0**	**79.4**	**56.7**	47.8

Table 4: Performance with different backbone LMs on unsupervised commonsense QA task.

tion task with standard evaluation metrics including MRR (Mean Reciprocal Rank) and Hits@10 to evalute the inductive CSKG completion models.

Benchmarks: In our experiments, following Wang et al. (2021), we use the inductive split of CN-82K and ATOMIC, where at least one of the entities in knowledge triplets of the testing sets is not present in the training set.

Baselines: We compare with ConvE (Dettmers et al., 2018), RotatE (Sun et al., 2019), Malaviya (Malaviya et al., 2020), InductivE (Wang et al., 2021) and MICO (Su et al., 2022).

Main Results: By training LMs with hard negative triplets and expanding the knowledge triplet with the potential missing alternatives on CSKGs, our method is able to generate superior commonsense knowledge representation, leading to the improved generalizability to unseen entities.

Table 2 shows the results of the inductive CSKG completion. Our method performs better on ATOMIC while remains comparable on ConceptNet. Previous entity embedding based methods by utilizing the existing entity links, such as ConvE (Dettmers et al., 2018) and RotatE (Sun et al., 2019), perform worse when it comes to the disconnected entities. For the graph neural network (GNN) based methods, such as Malaviya (Malaviya et al., 2020) and InductivE (Wang et al., 2021), by utilizing PLMs to initialize the entity embedding, the proposed GNNs trained on sampled subgraphs can significantly improve the generalizability on ConceptNet. However, the CSKGs are highly sparse and can be disconnected, the GNN-based methods could be failed when such a subgraph structure is not available (Franceschi et al., 2019).

In contrast, our method focuses on learning a relation-aware commonsense representation for each entity without relying on the graph structure. Same as MICO (Su et al., 2022), our method achieves better performance on ATOMIC while otherwise on ConceptNet compared with InductivE, one of the possible reasons could be the average length of the entity description in ATOMIC (6.12 words) is longer than that in ConceptNet (3.93 words). Longer sequences could enhance the PLMs to learn more accurate contextual representation for entity nodes. Compared with MICO, our method performs slightly worse on ConceptNet, one possible explanation is that more false negatives are introduced due to the hard negative sampling and positive set expansion.

5 Analysis

Ablation Study To further investigate what factors contribute to the performance gains, we conduct an ablation study by removing the step of hard negative sampling (HNS) and positive set expansion (PSE). Table 3 shows the results of ablation study on unsupervised CSQA task. Overall, when HNS or PSE is removed, the performance decreases on SIQA and CSQA whenever the model is trained with either ConceptNet or ATOMIC. Specifically, compared to the base model, training without HNS significantly hurts the performance by 2.6% and 0.7% on SIQA, which proves that hard negatives are effective in the existing contrastive learning instead of using in-batch negatives only. Meanwhile, removing PSE also degrades the performance most time, which shows that recovering the potential links between the head entity and the tail entity candidate by PSE contributes to learning superior commonsense-aware knowledge representation. However, removing PSE does not affect the accuracy much even can improve the performance slightly, which may be because that introducing PSE also incurs more false negatives in training.

Power of Scale We empirically test the influence of increasing the backbone LM size affecting the performance of the proposed model. Table 4 shows the results of different backbone LMs on unsupervised commonsense QA task. Overall, our method broadly benefits from backbone LM size increase. In addition, it conveys the same pattern as Table 1. ATOMIC benefits more for both COPA and SIQA, while ConceptNet is more helpful for CSQA.

6 Related Work

Contrastive Learning for NLP Contrastive learning has been applied into many NLP tasks. Such as, contrastive self-supervised objectives for text classification task (Fang et al., 2020; Kachuee et al., 2020); multi-view contrastive learning for dense encoder in open domain question answering (Karpukhin et al., 2020); sentence representation transfer with efficient contrastive framework (Yan et al., 2021; Gao et al., 2021). Among the works applying contrastive learning for NLP, Zhang and Stratos (2021) considered the importance of the hard negatives and proposed to combine hard negatives with appropriate score functions to improve the performance of zero-shot entity linking task. In this work, we propose to enhance contrastive learning with hard negative sampling for commonsense-aware knowledge representation task.

Unsupervised Commonsense Question Answering For the task of unsupervised CSQA, the vanilla PLMs can achieve moderate performance on most tasks. Furthermore, there are several methods generating intermediate outputs first by PLMs without relying on external CSKGs, such as SEQA (Niu et al., 2021), self-talk (Shwartz et al., 2020) and Dou (Dou and Peng, 2022). Some models incorporate CSKGs, including KTL (Banerjee and Baral, 2020), DynaGen (Bosselut et al., 2021), NLI-LM (Huang et al., 2021) and MICO (Su et al., 2022). Recently, a few methods prompt the large LMs to generate relevant knowledge given few-shot human annotations, including GKP (Liu et al., 2022) and TSGP (Sun et al., 2022). In this paper, we improve the commonsense knowledge representation by the sequence pairs synthesized CSKGs.

Commonsense Knowledge Graph Completion Existing KG completion methods can be adapted for CSKG completion, such as, ConvE (Dettmers et al., 2018) and RotatE (Sun et al., 2019) learn entity embeddings by the relation links between entity nodes. However, many entity nodes in CSKGs referring to the same concept are stored as distinct ones due to their free-form texts, resulting in larger and sparser graphs. To mitigate this issue, methods such as Malaviya (Malaviya et al., 2020) and InductivE (Wang et al., 2021), propose various graph neural network modules with the embeddings initialized from PLMs and focus on learn latent subgraph structures. Without leveraging graph structure, we also focus on the relation-aware knowledge representation with the free-form sequence pairs from CSKGS (Su et al., 2022).

7 Conclusion

In this paper, we propose to enhance the contrastive learning framework to fine-tune PLMs over CSKGs more effectively. Specifically, our method is divided into three steps: hard negative set sampling, positive set expansion and contrastive knowledge fine-tuning. We conduct extensive experiments on several unsupervised CSQA tasks and inductive CSKG completion with two widely used CSKGs, ConceptNet and ATOMIC. The performance gains demonstrate its effectiveness.

Limitations

First, in this paper, we focus on the commonsense knowledge representation learned on the synthesized sequence pairs from a given CSKG. However, the synthesized sequence pairs are missing contexts which may be indispensable for decision-making for some circumstances. Second, we propose to sample hard negatives during training instead of merely utilizing the in-batch negatives, which increases the memory footprint and computational costs. Third, we only focus on learning a relation-aware commonsense knowledge representation from the synthesized sequence pairs, while the subgraph structure of each entity node is also important for more fine-grained representation learning.

References

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162, Online. Association for Computational Linguistics.

Pratyay Banerjee, Swaroop Mishra, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2021. Commonsense reasoning with implicit knowledge in natural language. In *3rd Conference on Automated Knowledge Base Construction.*

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI).*

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi.

2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Ruben Branco, António Branco, Joao Rodrigues, and Joao Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521.

Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. 2016. Hard negative mining for metric learning based zero-shot classification. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 524–531. Springer.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

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*.

Zi-Yi Dou and Nanyun Peng. 2022. Zero-shot commonsense question answering with cloze translation and consistency optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10572–10580.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pages 1972–1982. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Canming Huang, Weinan He, and Yongmei Liu. 2021. Improving unsupervised commonsense reasoning using knowledge-enabled natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4875–4885, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Jinhao Ju, Deqing Yang, and Jingping Liu. 2022. Commonsense knowledge base completion with relational graph attention network and pre-trained language model. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4104–4108.

Mohammad Kachuee, Hao Yuan, Young-Bum Kim, and Sungjin Lee. 2020. Self-supervised contrastive learning for efficient user satisfaction prediction in conversational agents. *arXiv preprint arXiv:2010.11230*.

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 preprint arXiv:2004.04906*.

Yu Jin Kim, Beong-woo Kwak, Youngwook Kim, Reinald Kim Amplayo, Seung-won Hwang, and Jinyoung Yeo. 2022. Modularized transfer learning with multiple knowledge graphs for zero-shot commonsense reasoning. *arXiv preprint arXiv:2206.03715*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

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*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *35th AAAI Conference on Artificial Intelligence*.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2925–2933.

Yilin Niu, Fei Huang, Jiaming Liang, Wenkai Chen, Xiaoyan Zhu, and Minlie Huang. 2021. A semantic-based method for unsupervised commonsense question answering. 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 3037–3049, Online. Association for Computational Linguistics.

Bo Ouyang, Wenbing Huang, Runfa Chen, Zhixing Tan, Yang Liu, Maosong Sun, and Jihong Zhu. 2021. Knowledge representation learning with contrastive completion coding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3061–3073, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Miao Peng, Ben Liu, Qianqian Xie, Wenjie Xu, Hua Wang, and Min Peng. 2022. Smile: Schema-augmented multi-level contrastive learning for knowledge graph link prediction. *arXiv preprint arXiv:2210.04870*.

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.

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.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, pages 161–176.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ying Su, Zihao Wang, Tianqing Fang, Hongming Zhang, Yangqiu Song, and Tong Zhang. 2022. MICO: A multi-alternative contrastive learning framework for commonsense knowledge representation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1339–1351, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yueqing Sun, Yu Zhang, Le Qi, and Qi Shi. 2022. TSGP: Two-stage generative prompting for unsupervised commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 968–980, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Bin Wang, Guangtao Wang, Jing Huang, Jiaxuan You, Jure Leskovec, and C-C Jay Kuo. 2021. Inductive learning on commonsense knowledge graph completion. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Lihui Zhang and Ruifan Li. 2022. Ke-gcl: Knowledge enhanced graph contrastive learning for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 76–87.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. *arXiv preprint arXiv:2104.06245*.

A Details of CSKGs

A.1 CSKGs

Our experiments rely on two representative CSKGs, ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019a).

ConceptNet. ConceptNet focuses on taxonomic, lexical and physical commonsense knowledge, describing the relation between a conceptual entity with another entity. Li et al. (2016) first introduced CN-100K which contains Open Mind Common Sense entries in the ConceptNet5 knowledge base (Speer and Havasi, 2013) to separate true and false triplets. However, the data split ratio of CN-100K is biased. In view of this issue, we use the new data split CN-82K proposed in (Wang et al., 2021) that is uniformly sampled.

ATOMIC. ATOMIC is an event-centric knowledge base, which contains everyday commonsense knowledge organized as nine typed *if-then* relations, e.g. xIntent, xWant. It focuses on different aspects of an event, such as social effect, mental states and causes. Following previous work, we use CN-82K and ATOMIC in our experiments (Wang et al., 2021; Su et al., 2022). The statistics are shown in Table 7.

A.2 Templates for Relation

Table 5 and Table 6 show the template for relation used for ATOMIC and ConceptNet, we adopted the version from InductivE [1].

A.3 Evaluation Benchmarks for Unsupervised CSQA

We evaluate our framework on commonsense question answering datasets, COPA (Roemmele et al., 2011), SIQA (Sap et al., 2019b) and CSQA (Talmor et al., 2019). We evaluate on both the dev and test splits unless the test split is hidden. The label information is only used for the final accuracy calculation.

COPA (Roemmele et al., 2011) COPA is a two-alternative commonsense causal reasoning dataset, where one alternative is more plausible than the other. We replace the term *cause* with *The cause for it was that* and *effect* with *As a result*, as in previous work (Su et al., 2022).[2]

[1]https://github.com/BinWang28/InductivE

[2]Please refer to Su et al. (2022) for more details.

Relation	rel template
xAttr	PersonX is seen as
xEffect	as a result, PersonX will
xWant	as a result, PersonX wants
xNeed	but before, PersonX needed
xReact	as a result, PersonX feels
xIntent	because PersonX wanted
oEffect	as a result, PersonY or others will
oReact	as a result, PersonY or others feel
oWant	as a result, PersonY or others want
xAttr rev	"PersonX is seen as", "because PersonX"
xEffect rev	"PersonX will", "because PersonX"
xWant rev	"PersonX wants", "because PersonX"
xNeed rev	"PersonX needs", "as a result PersonX"
xReact rev	"PersonX feels", "because PersonX"
xIntent rev	"PersonX wanted", "as a result PersonX"
oEffect rev	"PersonY or others will", "because PersonX"
oReact rev	"PersonY or others feel", "because PersonX"
oWant rev	"PersonY or others want", "because PersonX"

Table 5: Relation types and relation substitute templates from ATOMIC. *rev* mean reverse relation.

SIQA (Sap et al., 2019b) SIQA is three-choice dataset for testing social commonsense knowledge. Questions are built upon ATOMIC, focusing on social interactions about people's actions and their social implications.

CSQA (Talmor et al., 2019) CSQA is collected based on ConceptNet. Each question explores the potential taxonomic or physical commonsense relationships between entities and has five crowd-sourced candidate answers.

B Experimental Settings

We mainly run our experiments with RoBERTa-Large (Liu et al., 2019), which consists of 355M parameters. Our experiments are conducted with a A100 GPU. The running time of each experiment is about 5 10 hours. The results are averaged by three experiments.

Relation	relation templates
AtLocation	located or found at or in or on
CapableOf	is or are capable of
NotCapableOf	is not or are not capable of
Causes	causes
CausesDesire	makes someone want
CreatedBy	is created by
DefinedAs	is defined as
DesireOf	desires
Desires	desires
NotDesires	do not desire
HasA	has, possesses, or contains
HasFirstSubevent	begins with the event or action
HasLastSubevent	ends with the event or action
HasPrerequisite	to do this, one requires
HasProperty	can be characterized by being or having
InstanceOf	is an example or instance of
IsA	is a
MadeOf	is made of
MotivatedByGoal	is a step towards accomplishing the goal
PartOf	is a part of
ReceivesAction	can receive or be affected by the action
SymbolOf	is a symbol of
UsedFor	used for
LocatedNear	is located near
RelatedTo	is related to
InheritsFrom	inherits from
LocationOfAction	is acted at the location of
HasPainIntensity	causes pain intensity of
AtLocation rev	is the position of
CapableOf rev	is a skill of
NotCapableOf rev	is not a skill of
Causes rev	because
CausesDesire rev	because
CreatedBy rev	create
DefinedAs rev	is known as
DesireOf rev	is desired by
Desires rev	is desired by
NotDesires rev	is not desired by
HasA rev	is possessed by
HasFirstSubevent rev	is the beginning of
HasLastSubevent rev	is the end of
HasPrerequisite rev	is the prerequisite of
HasProperty rev	is the property of
InstanceOf rev	include
IsA inversed	includes
MadeOf rev	make up of
MotivatedByGoal rev	motivate
PartOf rev	include
ReceivesAction rev	affect
SymbolOf rev	can be represented by
UsedFor rev	could make use of
LocatedNear rev	is located near
RelatedTo inversed	is related to
InheritsFrom rev	hands down to
LocationOfAction rev	is the location for acting
HasPainIntensity rev	is the pain intensity caused by

Table 6: Relation types and relation substitute templates from ConceptNet. *rev* mean reverse relation.

Dataset	Entities	Relations	Train Edges	Valid Edges	Test Edges	Avg. In-Degree
ConceptNet	78,334	34	81,920	9,795	9,796	1.31
ATOMIC	304,388	9	610,536	24,355	24,486	2.58

Table 7: Distribution of train, valid, and test edges from CN-82K and ATOMIC. Avg. In-Degree is the average number of tail entity connected to head entity.

Author Index