INLG 2024

The 17th International Natural Language Generation Conference: System Demonstrations

Proceedings of the System Demonstrations

September 23 - 27, 2024

Order copies of this and other ACL proceedings from:

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

ISBN 979-8-89176-123-0

Preface

We are excited to present the Proceedings of the 17th International Natural Language Generation Conference (INLG 2024). This year's INLG takes place from September 23-27 in Tokyo, Japan and is organized by the National Institute of Advanced Industrial Science and Technology. We would like to thank the local organizing team led by Tatsuya Ishigaki; the conference would not be possible without their dedication and hard work.

The INLG conference is the main international forum for the presentation and discussion of research on Natural Language Generation (NLG). This year, we received 98 conference submissions (including 2 from ARR) and 7 demo paper submissions. After a peer review process, 38 long papers, 19 short papers, and 6 demos were accepted to the conference and are included in these proceedings. The accepted papers showcase the breadth of NLG research, including work on applications, such data-to-text tasks, machine translation, and summarization; language model evaluation; and many other topics of interest to the NLG community. We thank Chung-Chi Chen for serving as Publication Chair and preparing these proceedings.

We are also excited to present four keynotes, which will discuss enhancing reasoning capabilities in NLG systems, applications of NLG to creative writing, evaluation of language generation, and embodied NLG for autonomous robots. The keynote speakers are:

- Yulan He, King's College London, UK
- Mark Riedl, Georgia Institute of Technology, USA
- Kees van Deemter, Utrecht University, the Netherlands
- Koichiro Yoshino, Tokyo Institute of Technology, Japan

For the second year, INLG is hosting a Generation Challenge, a track of the main conference focused on developing shared tasks for NLG. The track is chaired by Simon Mille and Miruna Clinciu. This year, there are three challenges: long story generation, visually grounded story generation, and the Generation, Evaluation, and Metrics (GEM) benchmark.

Two workshops are co-located with the main conference: the 2nd Workshop on Practical LLM-assisted Data-to-Text Generation and the 2nd Workshop of AI Werewolf and Dialog System. INLG is also hosting a tutorial on Human Evaluation of NLP System Quality. We also thank Jing Li for serving as Workshop Chair for the conference.

Finally, would like to thank our generous sponsors:

- Gold sponsors: Denso IT Library and Fast Accounting Co., Ltd.
- Silver sponsors: Stockmark Inc., Recruit Co., Ltd., and the Artificial Intelligence Research Center (AIRC).
- Bronze sponsors: Association for Natural Language Processing

We would also like to express our gratitude to the Area Chairs and Program Committee members for their reviewing contributions, and to the SIGGEN representatives Raquel Hervás and Emiel van Miltenburg for sharing their expertise.

Your INLG 2024 program chairs,
Saad Mahamood (lead), Nguyen Le Minh, and Daphne Ippolito

Organizing Committee

Program Chairs

Saad Mahamood (lead), trivago N.V.
Nguyen Le Minh, Japan Advanced Institute of Science and Technology
Daphne Ippolito, Carnegie Mellon University

Generation Challenge Chairs

Simon Mille, ADAPT Research Centre, Dublin City University, Ireland
Miruna Clinciu, Edinburgh Centre of Robotics

Local Organization Committee

Tatsuya Ishigaki (lead), National Institute of Advanced Industrial Science and Technology
Ayana Niwa, , Recruit Co., Ltd. / Megagon Labs
Takashi Yamamura, Yamagata Universit
Shun Tanaka, JX PRESS Corporation
Yumi Hamazano, Hitachi, Ltd.
Toshiki Kawamoto, Amazon
Takato Yamazaki LY Corp. / SB Intuitions Corp.
Hiroya Takamura, National Institute of Advanced Industrial Science and Technology
Ichiro Kobayashi, Ochanomizu University

SIGGEN Executives

Raquel Hervás (University Complutense of Madrid, Spain)
Emielvan Miltenburg (Tilburg University, the Netherlands)

Publication Chair

Chung-Chi Chen (National Institute of Advanced Industrial Science and Technology, Japan)

Sponsor Chair

Ayana Niwa, Recruit Co., Ltd. / Megagon Labs

Area Chairs

Albert Gatt, Utrecht University
Chris van der Lee, Tilburg University
Fahime Same, trivago N.V.
João Sedoc, New York University
Michael White, Ohio State University
Natalie Schluter, Apple
Ondrej Dusek, Charles University
Rudali Huidrom, ADAPT Centre
Samira Shaikh, University of North Carolina
Suma Bhat, University of Illinois at Urbana-Champaign
Wei-Yun Ma, Academia Sinica

Program Committee

Adarsa Sivaprasad, University of Aberdeen
Alberto Bugarín-Diz, University of Santiago de Compostela
Aleksandre Maskharashvili, Ohio State University
Alessandro Mazzei, University of Turin
Alyssa Allen, Ohio State University

Anastasia Shimorina, Orange
Antonio Valerio Miceli Barone, University of Edinburgh
Antonis Antoniades, University of California, Santa Barbara
Anya Belz, Adapt Centre
Asad Sayeed, University of Gothenburg
Ashley Lewis, Ohio State University
Balaji Vasan Srinivasan, Adobe
Bohao Yang, University of Sheffield
Brian Davis, Adapt Centre
C. Maria Keet, University of Cape Town
Chris van der Lee, Tilburg University
Christina Niklaus, University of St. Gallen
Craig Thomson, University of Aberdeen
Daniel Braun, University of Twente
Daniel Paiva, Arria NLG
Daniel Sanchez, University of Granada
David M. Howcroft, Edinburgh Napier University
David McDonald, Smart Information Flow Technologies
Di Wang, King Abdullah University of Science and Technology
Eduardo Calò, Utrecht University
Ehud Reiter, University of Aberdeen
Elizabeth Clark, Google
Emiel Krahmer, Tilburg University
Emiel van Miltenburg, Tilburg University
Gonzalo Mendez, Complutense University of Madrid
Gordon Briggs, U.S. Naval Research Laboratory
Guanyi Chen, Utrecht University
Guy Lapalme, University of Montral
Hiroya Takamura, National Institute of Advanced Industrial Science and Technology (AIST)
Hugo Contant, Carnegie Mellon University
Ingrid Zukerman, Monash University
Jan Trienes, University of Marburg
Jennifer Biggs, Defence Science and Technology Group
Judith Sieker, University of Bielefeld
Kathleen McCoy, University of Delaware
Kees van Deemter, Universiteit Utrecht
Kei Harada, University of Electro-Communications
Kim Gerdes, Université Paris Saclay
Kristina Striegnitz, Union College
Lara Martin, University of Maryland
Lea Krause, Vrije Universiteit
Maciej Zembrzuski, Huawei
Maja Popović, IU International University of Applied Sciences
Maja Stahl, Leibniz Universität Hannover
Marc Tanti, University of Malta
Mariet Theune, University of Twente
Mark Steedman, University of Edinburgh
Martijn Goudbeek, Tilburg University
Mary-Jane Antia, University of Capetown
Mayank Jobanputra, Saarland University
Michela Lorandi, Dublin City University

Michimasa Inaba, University of Electro-Communications
Mihir Kale, Google
Mika Hämäläinen, Metropolia University of Applied Sciences
Miriam Anschütz, Technische Universität München
Miruna Clinciu, Edinburgh Centre of Robotics,
Nadjet Bouayad-Agha, Universitat Oberta de Catalunya,
Nikolai Ilinykh, Göteborgs Universitet
Nina Dethlefs University of Hull
Pablo Duboue
Patricia Schmidtova, Charles University
Paul Youssef, Phillips-Universität Marburg
Philipp Sadler, University of Potsdam
Qingyun Wang, University of Illinois
Raquel Hervas, Complutense University of Madrid
Reno Kriz, Johns Hopkins University
Robert Weißgraeber, AX Semantics
Rodrigo de Oliveira, IQVIA
Rudali Huidrom, Adapt Centre
Ryo Nagata, Hyogo University of Teacher Education
Sadid A. Hasan, Microsoft
Simeon Junker, Universität Bielefeld
Simon Mille, Adapy Centre
Simone Balloccu, Charles University
Sina Zarrieß, University of Bielefeld
Somayajulu Sripada, University of Aberdeen
Steffen Pauws, Tilburg University
Suraj Pandey
Symon Stevens-Guille, The Ohio State University
Thiago Castro Ferreira, Federal University of Minas Gerais
Toky Raboanary, University of Cape Town
Valerio Basile, University of Turin
Van Bach Nguyen, University of Marburg
Wanzheng Zhu, Google
Wei-Yun Ma, Academia Sinica
Wenhu Chen, University of Waterloo
Xiangru Tang, Yale University
Xinnuo Xu, University of Edinburgh
Yanzhou Pan
Yingjin Song, Utrecht University
Yinhe Zheng, Tsinghua University
Yizhi Li, University of Manchester
Yoshinobu Kano, Shizuoka University
Zola Mahlaza, University of Cape Town

Table of Contents

Be My Mate: Simulating Virtual Students for collaboration using Large Language Models

Sergi Solera-Monforte
Pablo Arnau-González
Miguel Arevalillo-Herráez
Universitat de València
{sergi.solera, pablo.arnau, miguel.arevalillo}@uv.es

Abstract

Advancements in machine learning, particularly Large Language Models (LLMs), offer new opportunities for enhancing education through personalized assistance. We introduce "Be My Mate," an agent that leverages LLMs to simulate virtual peer students in online collaborative education. The system includes a subscription module for real-time updates and a conversational module for generating supportive interactions. Key challenges include creating temporally realistic interactions and credible error generation. The initial demonstration shows promise in enhancing student engagement and learning outcomes.

1 Introduction

In recent years, advancements in machine learning have impacted the educational landscape, particularly with the emergence of Large Language Models (LLMs). These technologies offer new opportunities to enhance the learning experience and provide personalised assistance. However, the possibility of using LLMs to create a collaborative learning environment has not been explored. Currently, LLMs have passed the Turing test and it is difficult to tell LLM-generated text content apart from human-generated text. This, along with the emergent abilities in LLMs (Chang et al., 2024), shows the possibility of having LLM impersonate a peer student.

In the context of online collaborative education, it is often difficult to pair students with peers, or obtain a large enough pool of students to group individuals. For this reason, we propose the use of LLMs disguised as students to generate interaction with other students as their peers. This would enable benefits associated with collaborative learning, without the difficulties of pairing students.

However, the implementation of these systems is not straightforward. We offer a first demo of

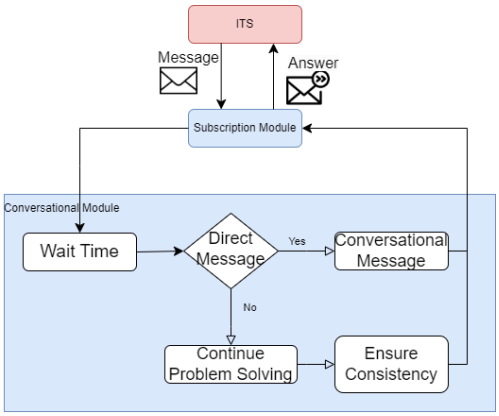


Figure 1: Processing Message Schema

Be My Mate[1], an initial attempt to simulate virtual students for collaboration using large language models. By leveraging the capabilities of LLMs, we aim to create virtual entities that can interact with human students, participate in group discussions, and contribute to collaborative projects.

2 Agent Implementation

We start from (Arevalillo-Herráez and Arnau, 2013), an existing system that supports the collaborative resolution of math word problems, while providing active feedback. The system utilises a WebSocket-based queue notification protocol, STOMP, for publishing updates about the state of the solution and the conversation of a given solution instance (room).

The proposed agent must subscribe to these updates to receive the current status and produce relevant contributions to the problem-solving. In order to do so, we split the system in two main modules, as shown in Figure 1:

Subscription module: Establishes the connection with the Intelligent Tutoring System (ITS) and subscribes for updates on the generated rooms. Every time a new room is created, the module will

[1] https://github.com/SPAM-research/BeMyMate

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 1–3
September 23–27, 2024.

listen to the user's inputs and modifications on the solution status (i.e. notes or equations) to propose new solutions or a conversational message to the student.

Conversational module: Its main goal is to provide relevant messages for supporting the student. The module has to decide whether to send a message to the student or propose a new letter, or equation. Consistency with the current context is ensured with two components that prepare, or discard, the output of the model to conform with the expected outputs in our system. In this direction, we check if the user has directly spoken to his peer (the agent), and therefore will be expecting a response, or, on the contrary, the system can freely propose new steps, such as letter assignation or equation definition. Alternatively, in case of a message to advance the resolution status, we must ensure the consistency of such message in terms of the expected message by the ITS. These validations aim to ensure the messages are credible. For example, ensuring that the provided equations are correctly structured for a mathematical context or that the letters used in the proposed equations by the agent have been previously defined. This validation must not prevent the agent from failing, but from making mistakes that may not be credible.

3 Technical and scientific challenges

During the analysis phase, we have identified temporisation as a main factor that must be addressed to convince alumni that they are interacting with peers. In the context of our ITS, humans tend to take more than a couple of seconds to write a chat message or propose a new definition. The time taken to write a message is influenced by a variety of factors, such as how advanced the problem is (at the beginning of a problem, humans try to understand it before writing the first message), familiarity with the system, age, or others. This time should be modelled and depends on the specifics of the system to be integrated in. At this stage and after manually inspecting a variety of values, we decided to include a uniform random wait time between 2 and 6 seconds for generating a message to the user, and between 5 and 10 seconds for generating the next step.

Another relevant factor that has been found relevant, is the type of mistakes the LLM can produce. In our context (an arithmetic, and algebra ITS) human mistakes are often related to the identification of relations between the problem quantities. The impersonated student should commit mistakes as if it were a regular student. This would create a perception of interacting with a peer. However, the errors should be credible. Moreover, although our system is capable of sending direct messages to the students, the agent should be able to decide to directly talk to the student to explain a concept or ask questions, as a peer would do.

Finally, regarding data privacy and collection, taking into account the sensitivity of educational-related data, authors should only capture the required samples to evaluate the system while preserving the student's anonymity. To this end, a new challenge arises for deciding if a student utterance is safe or not (directly leaks any personal data), correct it if possible, or directly reject it. Moreover, and appropiate storage scheme must be implemented to ensure that only individuals with research access can access the confidential data.

4 Conclusions

The demonstrated work[2] is a proof of concept of what we are aiming for in the Be My Mate agent. Although some aspects of our demo require further refinement, it demonstrates that LLM-based agents are a feasible option for providing collaborative learning environments. While significant progress must be made before agents can truly emulate humans and pass the Turing Test -such as timing the agent's interventions to enable a realistic conversational flow-, this demo highlights the potential of using LLMs to exploit the advantages of collaborative learning without the need for multiple students to be connected simultaneously. This first stone opens many research opportunities such as is addressing the bias these models might contain in the educational context. Moreover, the usefulness of these models, although theoretically sound must be empirically validated, either by studying the agent-student interaction by experts, in terms of learning compared to a Control Group, or by directly surveying the students with relevant tests.

In addition, the use of LLMs replacing real students can provide further benefits in this type of setting. In particular, it allows the system to rely on a single student model to make decisions, such as selecting the next activity, determining the optimal level of scaffolding, and tailoring the content and

[2]Video Available at: https://mmedia.uv.es/fbs?cmd=view&name=be_my_pal_demo.mp4

system behavior to the learner's needs and preferences.

Acknowledgments

Research supported by project CIGE/2023/63 funded by Generalitat Valenciana; TED2021-129485B-C42, funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR; and Generatitat Valenciana through grants CIAPOS/2022/163; ACIF/2021/439.

References

Miguel Arevalillo-Herráez and David Arnau. 2013. A hypergraph based framework for intelligent tutoring of algebraic reasoning. In *Artificial Intelligence in Education*, pages 512–521, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

MTSwitch: A Web-based System for Translation between Molecules and Texts

Nijia Han[1], **Zimu Wang**[1], **Yuqi Wang**[1], **Haiyang Zhang**[1], **Daiyun Huang**[2], **Wei Wang**[1,†]
[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University
[2]XJTLU Wisdom Lake Academy of Pharmacy, Xi'an Jiaotong-Liverpool University
{Nijia.Han23,Zimu.Wang19,Yuqi.Wang17}@student.xjtlu.edu.cn
{Haiyang.Zhang,Daiyun.Huang,Wei.Wang03}@xjtlu.edu.cn

Abstract

We introduce MTSwitch, a web-based system for the bidirectional translation between molecules and texts, leveraging various large language models (LLMs). It supports two crucial tasks, including molecule captioning (explaining the properties of a molecule) and molecule generation (designing a molecule based on specific properties). To the best of our knowledge, MTSwitch is currently the first accessible system that allows users to translate between molecular representations and descriptive text contents. The system and a screencast can be found in https://github.com/hanninaa/MTSwitch.

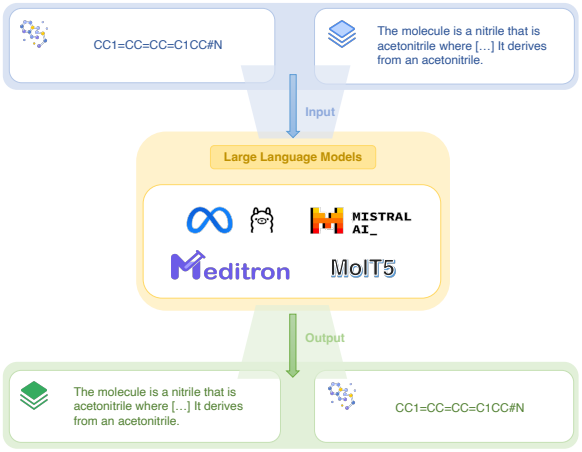


Figure 1: System overview of the MTSwitch system.

1 Introduction

The advent of large language models (LLMs) has significantly transformed the landscape of natural language processing (NLP) (Peng et al., 2023; OpenAI et al., 2024). Their superior language understanding, generation capabilities, and versatility have propelled their utility beyond conversations, now extending into various domains, including bio-medicine (Wang et al., 2023) and molecular chemistry (Liao et al., 2024). This emerging trend, marked by many novel methods being proposed, highlights a promising new research direction (Zeng et al., 2022; Edwards et al., 2022).

Despite the progress made, using the existing models requires familiarisation with invoking the advanced models, which undoubtedly increases the burden on researchers who are not specialists in this field. Consequently, designing an intuitive and user-friendly system for translating molecules and texts is imperative. In this paper, we design the first web-based system, named MTSwitch (Molecule-Text Switch), to translate between texts and molecules. As shown in Figure 1, MTSwitch can translate molecules in SMILES (Simplified Molecular Input Line Entry System) format along with their captions, based on the translation task, the selected model, and the user's input. In this system, we incorporate four models: the trained MolT5 and Meditron from Edwards et al. (2024), Llama 3 (Touvron et al., 2023), and Mistral (Jiang et al., 2023) trained with a subset of L+M-24 (Edwards et al., 2024). Our system empowers users with direct web access, facilitating seamless integration into education and molecular design.

2 System Overview

MTSwitch offers a user-friendly platform for translation between molecules in SMILES format and natural language, whose web interface is structured as two areas for input and output. As shown in Figure 1, users are prompted to first select an appropriate model (e.g. MolT5 and Meditron) and define their intended task when they access the platform, and then enter their input (either a SMILES notation or a natural language text) in the text box on the left-hand side. Upon submission, the system processes the user input, selects and executes the corresponding model for either molecule captioning or molecule generation, and exhibits the model output in the text box on the right-hand side.

†Corresponding author.

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 4–6
September 23–27, 2024.

Model	ROGUE-1↑	ROGUE-2↑	ROGUE-L↑	BLEU-2↑	BLEU-4↑	Meteor↑
Llama 3	66.9	50.1	50.3	61.2	44.1	60.6
Mistral	74.5	55.5	53.6	70.9	51.0	69.0
Meditron	**78.8**	**58.3**	**56.5**	**78.1**	**56.7**	**74.7**
MolT5	75.9	55.4	53.7	75.7	54.2	71.5

Table 1: Experimental results of the molecule captioning task in MTSwitch system with different LLMs, in which the best result for each metric is highlighted in **bold**.

Model	BLEU↑	Levenshtein↓	Validity↑	Uniqueness↑	MACCS↑	RDK↑	Morgan↑
Llama 3	63.6	56.8	92.2	81.2	64.9	57.1	42.1
Mistral	68.5	49.5	91.4	82.6	69.4	62.5	46.1
Meditron	68.4	45.8	**98.8**	97.9	**75.3**	**67.1**	47.8
MolT5	**71.5**	**42.1**	94.5	**99.5**	73.1	65.9	**48.5**

Table 2: Experimental results of the molecule generation task in MTSwitch system with different LLMs, in which the best result for each metric is highlighted in **bold**.

We provide comprehensive model support for the translation between molecules and texts. First, we employ two LLMs pre-trained on the L+M-24 dataset (Edwards et al., 2022), which is a large-scale dataset designed for translating molecules and texts: Meditron and MolT5, which are obtained from the Hugging Face repository[1]. We further fine-tune two state-of-the-art LLMs, Llama 3 (Touvron et al., 2023) and Mistral (Jiang et al., 2023), using a subset of the L+M-24 dataset. During our fine-tuning process, we employ the same instruction and hyperparameters intended for the fine-tuned Meditron model.

This system bridges the gap between complex molecule structures and their textual descriptions, thereby facilitating the understanding and communication within chemistry-related disciplines. By providing an intuitive interface for such translations, MTSwitch aims to reduce the cognitive efforts required to interpret molecular structures or craft SMILES notation from descriptive texts.

3 Evaluation

To evaluate the performance of MTSwitch, we adopted the evaluation metrics introduced by Edwards et al. (2022), which had taken both semantic similarity in natural languages and Fingerprint Tanimoto Similarity (FTS) for molecules into account. ROUGE, BLEU, and METEOR are metrics for evaluating generation quality by measuring n-gram overlap, precision with brevity penalties, and considering synonymy and stemming, respectively. MACCS, RDK, and Morgan are types of molecular fingerprints that represent molecular structures as binary vectors, with MACCS using predefined structural keys, RDK encoding specific molecular features, and Morgan (ECFP) generating circular fingerprints by expanding atom environments. We sampled a subset from the validation set of the L+M-24 dataset and assessed the performance of all models across the two tasks.

Tables 1 and 2 highlight the model performance in molecular captioning and generation, respectively. Meditron performs exceptionally well in most captioning metrics, particularly ROUGE-2 and BLEU-2. While Llama 3 performed well in ROUGE-2 and ROUGE-L, it underperformed in BLEU-4 and Meteor, indicating its potential coherence and lexical diversity issues in specific text generation tasks. For molecular generation, MolT5 and Meditron demonstrated superior performance across key indicators, highlighting their potential for advanced molecular design and synthesis prediction applications. The performances of these two models in our system closely aligned with their originally reported results (Edwards et al., 2024).

4 Conclusion

We introduced MTSwitch, a novel and user-friendly web-based system for bidirectional translation between molecules and texts. Leveraging LLMs, our system demonstrated superior performance, offering users an efficient platform for translating between molecular representations and textual descriptions seamlessly. In the future, we plan to incorporate more models and tasks into the system to make it more comprehensive and effective.

[1] https://huggingface.co/language-plus-molecules

References

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022. Translation between molecules and natural language. In *Proceedings of EMNLP*, pages 375–413.

Carl Edwards, Qingyun Wang, Lawrence Zhao, and Heng Ji. 2024. *L+M-24*: Building a dataset for language + molecules @ acl 2024. *Preprint*, arXiv:2403.00791.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Chang Liao, Yemin Yu, Yu Mei, and Ying Wei. 2024. From words to molecules: A survey of large language models in chemistry. *Preprint*, arXiv:2402.01439.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2023. When does in-context learning fall short and why? a study on specification-heavy tasks. *Preprint*, arXiv:2311.08993.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Yuqi Wang, Zeqiang Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, and Suparna De. 2023. Zero-shot medical information retrieval via knowledge graph embedding. In *Proceedings of IOTBDH*, pages 29–40.

Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2022. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1).

VideoRAG: Scaling the context size and relevance for video question-answering

Shivprasad Sagare, Prashant Ullegaddi, Nachiketh K S,
Navnith R, Kinshuk Sarabhai, Rajeshkumar SA
PhroneticAI
Correspondence: shivprasad.sagare@phronetic.ai, rajesh.kumar@phronetic.ai

Abstract

Recent advancements have led to the adaptation of several multimodal large language models (LLMs) for critical video-related use cases, particularly in Video Question-Answering (QA). However, most of the previous models sample only a limited number of frames from video due to the context size limit of backbone LLM. Another approach of applying temporal pooling to compress multiple frames, is also shown to saturate and does not scale well. These limitations cause videoQA on long videos to perform very poorly. To address this, we present VideoRAG, a system to utilize recently popularized Retrieval Augmented Generation (RAG) pipeline to select the top-k frames from video, relevant to the user query. We have observed a qualitative improvement in our experiments, indicating a promising direction to pursue. Additionally, our findings indicate that VideoRAG demonstrates superior performance when addressing needle-in-the-haystack questions in long videos. Our extensible system allows for trying multiple strategies for indexing, ranking, and adding QA models.

1 Introduction

In the realm of video-based language models (Video-LLMs), the ability to accurately answer questions about long-form video content remains a significant challenge. VideoRAG, a novel system introduced by researchers, aims to address this issue by retrieving more relevant and informative video frames to enhance the context for video-LLMs.

Several recent works adapt the pre-trained image-text multimodal LLMs to video data, by encoding multiple video frames into a sequence of features. However, this approach is limited by the trade-off between the number of frames and the computation cost. PLLaVA (Xu et al., 2024), a SOTA videoQA model, uniformly samples only 16 frames, which is bound to miss the key details in long videos.

Straightforward approach of averaging the spatial and temporal dimensions, as implemented in the VideoChatGPT system (Maaz et al., 2023), leads to the loss of substantial spatial information and fails to achieve optimal performance as the training dataset is scaled (Xu et al., 2024).

In contrast, VideoRAG employs a retrieval-augmented approach, where the system dynamically retrieves the most relevant video frames to supplement the input to the video-LLM. VideoRAG allows indexing of multiple video frames using multiple encoders, and searching the top-k most relevant frames given a query. We also release a human-annotated benchmark dataset along with our system, specially curated for long videos.

The evaluation of VideoRAG on question-answering tasks shows that it outperforms baseline video-LLMs that rely on direct frame averaging or fixed-frame sampling, particularly on long videos.

2 VideoRAG system

Our VideoRAG system leverages standard components of the Retrieval-Augmented Generation (RAG) pipeline and image encoding. It is platform-agnostic, supporting the addition of various strategies at each component level. The system architecture comprises six main components:

Video Processing We extract the candidate frames from the video, focusing on efficient and scalable processing to manage the memory overhead associated with storing frame information. Other aspects of the video are extracted as well, for example the audio signal, and the objects appearing the video. We also process the videos to extract chunks to track the activities happening during these chunks.

Video Encoding Entities from video, such as frames, chunks, and other extracted information

Our demo screencast is available at video link

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 7–8
September 23–27, 2024.

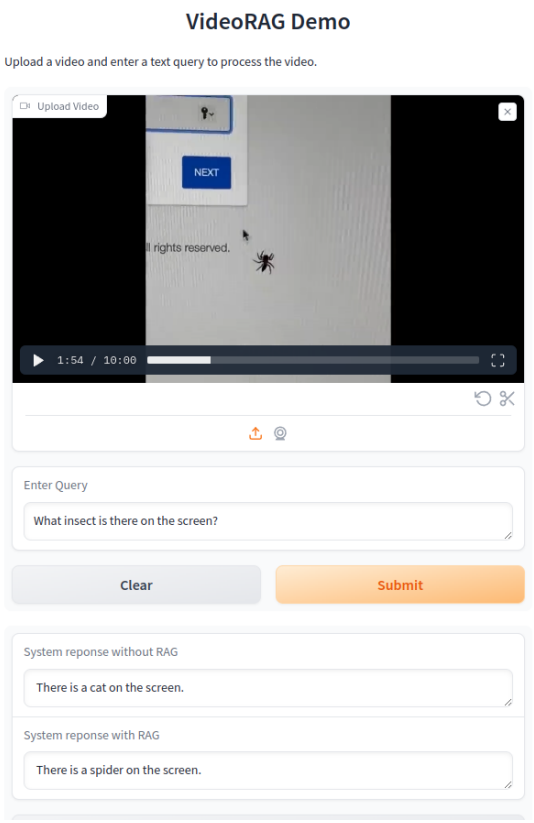


Figure 1: An example of VideoRAG.

is encoded into fixed-size vector semantic representations for further indexing. We explore multiple strategies to encode the videos intelligently, including the features like image frames, audio signal, and the objects appearing in the video. We rely on multiple strategies like SentenceTransformers (Reimers and Gurevych, 2019), (Li et al., 2022), (Zhai et al., 2023) for encoding the text query.

Indexing We employ a vector database, to store the frame embeddings effectively. This is a standard component in the RAG pipeline. These vector databases allow for efficient lookup algorithms for retrieval of top-k samples.

Query Encoding The text query is encoded similarly to the video encoding strategy to perform semantic similarity search over the index.

Retrieval and Ranking The vector database retrieves the top-k entities matching the query by performing efficient similarity search. We combine the results from multimodal indexes to generate the final ranking for the videos.

Generation with Video LLM Finally, we pass the selected entities and the text query to the video-text LLM model. We generate the answers using the state-of-the-art video-text LLM for this phase.

Metrics	Without RAG	With RAG
Correctness	2.00	**2.46 (+23%)**
Detail	2.45	**2.75 (+12%)**
Context	2.48	**2.94 (+18%)**

Table 1: Comparison of VideoRAG against traditional systems denotes substantial gains in accuracy and context while generating the answers. The scores are from LLM-based evaluation with the range of 1(worst) to 5(best).

3 Usage and Evaluation

VideoRAG is planned to be used for easy experimentation, and evaluation of long form video question-answering systems. Our system allows for easy extension to additional indexing mechanisms, ranking strategies, and videoQA models. We compare our VideoRAG outputs with that of non-RAG baseline systems. We curated a dataset of a few long videos and corresponding question-answer pairs. Table 1 showcases the quantitative evaluation results, using metrics and evaluation scripts from VideoChatGPT evaluation framework (Maaz et al., 2023). Our observations indicate that VideoRAG effectively minimizes hallucinations when addressing needle-in-the-haystack questions. We attribute this to its ability to retrieve relevant video entities in response to user queries.

References

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Preprint*, arXiv:2201.12086.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *Preprint*, arXiv:2306.05424.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava : Parameter-free llava extension from images to videos for video dense captioning. *Preprint*, arXiv:2404.16994.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

QCET: An Interactive Taxonomy of Quality Criteria for Comparable and Repeatable Evaluation of NLP Systems

Anya Belz, Simon Mille, Craig Thomson and **Rudali Huidrom**
DCU Natural Language Generation Research Group
ADAPT, Dublin City University, Dublin, Ireland
anya.belz@dcu.ie

Abstract

Four years on from two papers (Belz et al., 2020; Howcroft et al., 2020) that first called out lack of standardisation and comparability in quality criteria assessed in NLP system evaluations, researchers still use widely differing quality criteria names and definitions, meaning that it continues to be unclear when the same aspect of quality is being assessed in two evaluations. While normalised quality criteria were proposed at the time, the list was unwieldy and using it came with a steep learning curve. In this demo paper, our aim is to address these issues with an interactive taxonomy tool that enables quick perusal and selection of the standardised quality criteria, and provides decision support and examples of use at each node.

1 Introduction

Belz et al. (2020) and Howcroft et al. (2020) described a situation in NLP evaluation where over 200 different quality criteria (QC) names were in use, definitions were patchy, and it was impossible to tell if the same aspect of quality was being evaluated in different evaluations, resulting in low comparability and repeatability. Consider the following examples which share the same QC name, **Fluency**, but use very different definitions:

1. Yu et al. (2020): "judging the question fluency."
2. Van de Cruys (2020): "grammatical and syntactically well-formed."
3. Pan et al. (2020): "follows the grammar and accords with the correct logic."

The authors mapped the 200+ different QCs to a set of 71 standardised QC names and definitions. This implies that two thirds of the original set did not reflect actual differences between aspects of quality assessed, merely a lack of standardisation.

While highly cited, the work has found little practical application in that researchers continue to create new names and/or definitions for the aspects of quality they assess. We diagnose two reasons for this: (i) it is not straightforward to separate *what* is being assessed from *how* it is being assessed; and (ii) the 71 QCs were difficult to assimilate and use.

2 Disentangling the *What* from the *How*

What is being assessed refers to the specific aspect of quality (the QC) that an evaluation aims to assess. *How* refers to the way in which the QC is mapped to a specific measure that can be implemented in an evaluation method. The distinction matters, because there are many different ways in which the same quality criterion can be assessed (Belz et al., 2020). The different elements relate as follows (*evaluation mode*: intrinsic vs. extrinsic, subjective vs. objective and absolute vs. relative):

> Quality criterion + evaluation mode = evaluation measure;
> Evaluation measure + experimental design = evaluation method.

3 The Underlying QC Taxonomy

The root node is the single most general QC, Overall Quality. The next level down has the following three QC subclasses (Belz et al., 2020):

1. **Correctness**: The conditions under which outputs are maximally correct (hence of maximal quality) can be stated. E.g. for *Grammaticality*, outputs are (maximally) correct if they contain no grammatical errors.
2. **Goodness**: It cannot be stated when outputs are maximally good, only which of two is better/worse. E.g. for *Fluency*, even if an output contains no disfluencies, there may be other ways to improve its fluency.
3. **Features**: For a feature-type QC *X*, outputs are not generally better if they are more (or less) *X*. Depending on context, more *X* may be better or worse. E.g. *Funny* and *Entertaining* might be good in a narrative generator, but neither are appropriate in a nursing report generator.

At the next level, these three nodes split into subclasses that reflect whether outputs are assessed **in their own right**, **relative to the input**, or **relative to an external frame of reference** (e.g. comparison to a gold standard).

At a subsequent level, classes further split into the following three subclasses capturing which aspect of an output is being assessed:

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 9–12
September 23–27, 2024.

1. **Form**: The form of outputs alone is assessed, e.g. *Grammaticality* – a sentence can be grammatical yet wrong or nonsensical in terms of content.
2. **Content**: The content/meaning of outputs alone is assessed, e.g. *Meaning Preservation* – two sentences can have the same meaning, but differ in form.
3. **Both form and content**: Outputs are assessed as a whole. E.g. *Coherence* is a property of outputs as a whole, either form or meaning can detract from it.

The rest of the taxonomy consists of 70+ leaf and internal nodes; the following is an example of a complete branch from root to leaf node:

Overall quality of outputs → Correctness of outputs → Correctness of outputs in their own right → Correctness of outputs in their own right (Form) → Grammaticality.

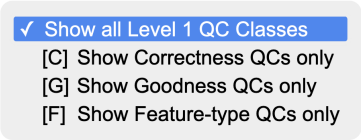
4 The QCET Tool

The Quality Criteria for Evaluations Taxonomy (QCET) tool has three core functionalities: (i) tree navigation via *show child nodes* (+) and *collapse subtree* (-); (ii) tree pruning to one of the subclasses in Section 3; (iii) viewing node details.

The starting interface (Appendix A) has instructions corresponding to Section 3 and the use cases below, followed by the three tree pruning pull-downs, and the taxonomy viewer, initially showing just the root node (*Quality of Outputs*).

The taxonomy tree can then be navigated by progressively showing child nodes and collapsing subtrees. Alternatively, to provide less comprehensive views, the taxonomy tree can be pruned in the following three ways:

1. **Prune by Level 1 QC classes**: show only one of the three Level 1 subtrees, corresponding to showing *Correctness* QCs, *Goodness* QCs, or *Feature-type* QCs only. Options are selected via the following pull-down:



2. **Prune by Level 2 QC classes**: show all and only subtrees rooted at *one* of the following: a QC evaluating outputs *In their Own Right*, a QC evaluating outputs *Relative to the Inputs*, or QC evaluating outputs *Relative to an External Frame of Reference*.
3. **Prune by Form/Content/Both QC classes**: show all and only subtrees rooted at *one* of the following: a *Form* QC, a *Content* QC, or a *Form and Content* QC.

The three pruning dimensions can be combined, so that e.g. selecting *Correctness QCs only* and *Form QCs only* at the same time will show just subtrees that are in both these classes.

Nodes. Each *QC node* has the following elements: (i) node ID, (ii) QC name; (iii) QC definition; (iv) suggested questions to put to the evaluators in a corresponding human evaluation of this criterion (a) in absolute and (b) relative evaluation mode; and (v) additional notes and information.

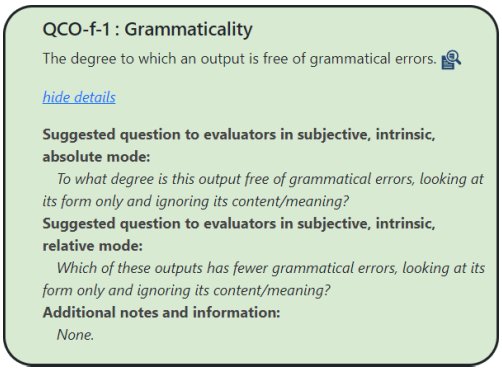


Figure 1: Example node with details shown.

The node ID traces the path from the root to the node, e.g. *QCO-f*: Quality→Correctness→In its own right→Form. Each node has a (+) button or a (-) button, standing for *show child nodes* and *collapse subtree*, respectively. Nodes by default show just the QC name and definition; this view can be expanded in situ, by clicking on *show details*, or by clicking on the magnifying class icon which opens the full view of the node details in a new tab.

See Figure 1 for an example of a non-expanded node viewed with details shown. Appendix B shows the expanded view of a node.

Use cases. We envisage two main uses: (i) Mapping previous evaluations to standardised QCs for comparability, and (ii) choosing a QC name and definition for new evaluations. In both cases, the default use mode is to peruse the taxonomy from the root node down until the right node is found (more details in Appendix C). For users more familiar with the taxonomy, the tree-pruning pull-downs offer a convenient way to reduce the search space.

Extensibility. The QCET tool is intended as an extensible resource, to which new leaf nodes can be added as needed. The tool includes an interface for proposing new nodes to be inserted at a specific parent node in the publicly shared version which the QCET team will review and, by default, add.

5 Conclusion

We have presented QCET,[1] a tool designed to make it easier to (i) identify which standardised QC is being assessed in an existing evaluation, and (ii) to select standard names and definitions to use in new evaluations, in order to increase comparability and repeatability of evaluation experiments overall.

[1] https://github.com/DCU-NLG/qcet

References

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

David Howcroft, Anya Belz, Miruna Clinciu, Dimitra Gkatzia, Sadid Hasan, Saad Mahamood, Simon Mille, Sashank Santhanam, Emiel van Miltenburg, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Natural Language Generation Conference*.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475, Online. Association for Computational Linguistics.

Tim Van de Cruys. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480.

Qian Yu, Lidong Bing, Qiong Zhang, Wai Lam, and Luo Si. 2020. Review-based question generation with adaptive instance transfer and augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 280–290, Online. Association for Computational Linguistics.

A QCET Starting Interface

Figure 2 shows the QCET starting interface. The user can view the instructions under the 5 headers at the top, before pruning the Taxonomy Tree with the dropdown menus and then expanding/collapsing nodes and their details on the tree itself.

B Single Node View

Figure 3 shows the details for node `QCO-f`; `Correctness of outputs in their own right (form)`. In addition to showing the definition, the Node Details show suggested questions for different evaluation modes. Also shown is the parent node `QCO` and the child nodes `QCO-f-1` and `QCO-f-2`.

Figure 4 shows details for the leaf node `QCO-f-1`; `Grammaticality`, the first child node of `QCO-f`.

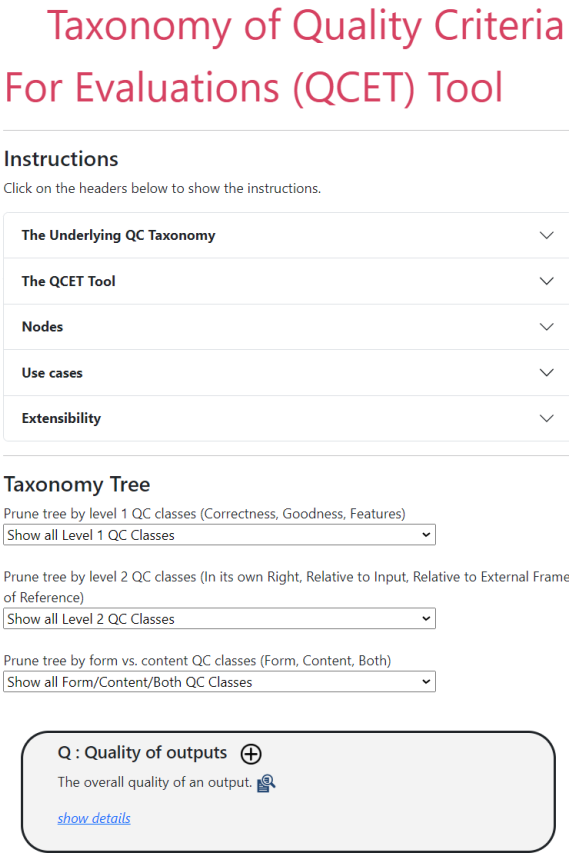


Figure 2: Example node in single view; Correctness of outputs in their own right (form).

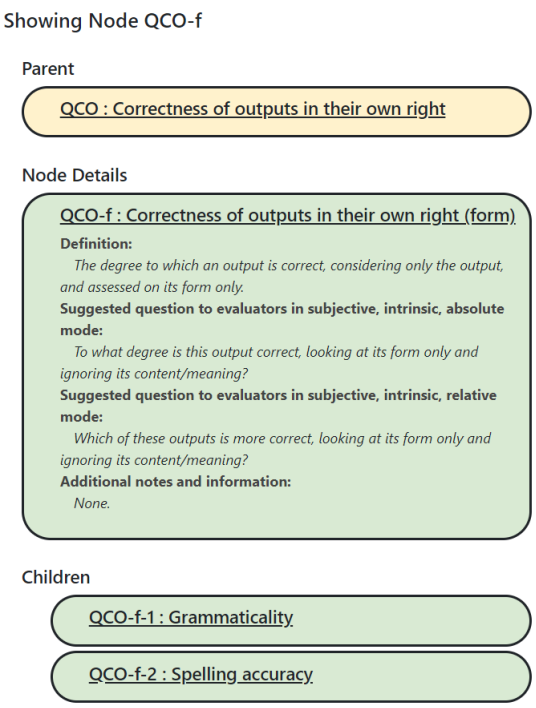


Figure 3: Example node in single view; Correctness of outputs in their own right (form).

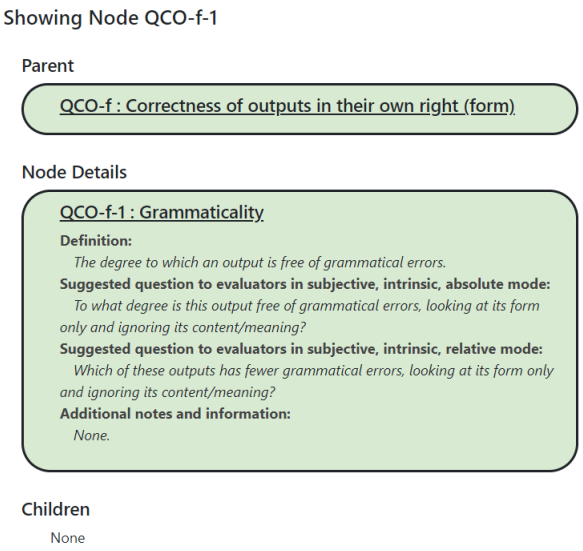


Figure 4: Example node in single view; Grammaticality (child node of QCO-f).

C Example Uses

Identifying standardised QCs in existing experiments. The first step is to locate all resources shared about a given experiment, then to identify (i) QC name, (ii) QC definition and (iii) the question and/or instructions put to evaluators. In many cases, (i)–(iii) are not completely aligned in which case (iii) takes priority as expressing what was actually evaluated.

A complicating factor is often that in the effort of explaining one QC, developers often introduce terms associated with other QCs, e.g. in the second Fluency definition at the start of the paper, Fluency is explained (only) in terms of grammaticality which introduces another QC. The third definition introduces two other QCs. We would argue that in the former case, one QC is being evaluated (Grammaticality), and in the latter case two (Grammaticality and Logicality). Neither assesses Fluency.

To arrive at these conclusions, armed with the information above, the taxonomy is perused via the QCET tool top down until the correct node(s) is/are reached (corresponding to the specific individual quality criteria above).

Designing new evaluation experiments. Once an initial idea has been formed about what aspect to assess in the new evaluation, the taxonomy is perused top down until the correct node(s) is/are reached. What is not supplied by the taxonomy is what explanations and/or instructions to issue to evaluators. Our recommendation would be to avoid paraphrasing the QC name and to rely more on examples for each response value. E.g. if a 5-point rating scale is used for Fluency, then provide a set of examples for each point on the scale. We are planning to add standardised questions for each node in the future.

factgenie: A Framework for Span-based Evaluation of Generated Texts

Zdeněk Kasner **Ondřej Plátek** **Patrícia Schmidtová**
Simone Balloccu **Ondřej Dušek**
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
${surname}@ufal.mff.cuni.cz

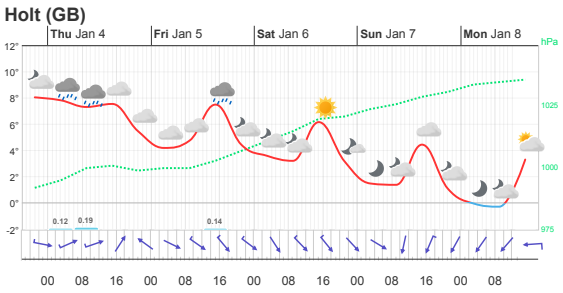
Abstract

We present factgenie: a framework for annotating and visualizing word spans in textual model outputs. Annotations can capture various span-based phenomena such as semantic inaccuracies or irrelevant text. With factgenie, the annotations can be collected both from human crowdworkers and large language models. Our framework consists of a web interface for data visualization and gathering text annotations, powered by an easily extensible codebase.[1]

1 Introduction

The fluency of texts generated by large language models (LLMs) is reaching the level of human-written texts. However, the texts generated by LLMs still contain various types of errors such as incorrect claims, claims not grounded in the input, or irrelevant statements. For precise and fine-grained evaluation of model outputs, it is necessary to identify these errors on the level of word spans. There are two major ways to collect the span annotations: using either human (Thomson and Reiter, 2020) or LLM-based annotators (Kocmi and Federmann, 2023; Kasner and Dušek, 2024).

None of the existing NLP error annotation platforms are suitable for gathering and visualizing word-level annotations from both human and LLM-based annotators. Some platforms are limited to specific tasks like machine translation (Klejch et al., 2015) and retrieval-augmented generation (ES et al., 2024). Other platforms are more flexible but allow either only human (Federmann, 2018; Nakayama et al., 2018) or only LLM (Dalvi et al., 2024) annotations. Systems supporting both annotation modalities typically include humans as post-editors only (Kim et al., 2024) and existing evaluation or visualization platforms require externally pre-annotated data (Trebuňa and Dušek, 2023; Masson et al., 2024; Fittschen et al., 2024).



(a) Custom visualization of the input data.

llama2

The weather in Holt, England, will be mostly cloudy with scattered clouds for the next five days. The temperature will range from 4.2 to 7.4 degrees Celsius, with a high of 6.1 degrees Celsius on January 7th. The wind speed will be around 2.5 to 4.6 meters per second, with gusts of up to 9.32 meters per second on January 4th. There will be light rain on January 4th, with a total of 0.12 millimeters of precipitation. The humidity will be around 77% to 97%.

(b) Annotated model output.

Figure 1: Elements from the factgenie user interface: (a) custom visualization of the input data, (b) the corresponding LLM output with span annotations. The highlight colors correspond to custom annotation categories defined for the annotation process (= incorrect fact, = fact not checkable, = misleading fact).

The lack of suitable tools for span-based error annotation motivated us to develop factgenie, a lightweight and customizable framework that enables collecting annotations from both humans and LLMs. Specifically, factgenie can be used both to (a) collect annotations from human workers through crowdsourcing services and (b) collect annotations by prompting an LLM through an API. Besides that, factgenie can be used for visualizing the input data and the corresponding model outputs.

The software design of factgenie targets researchers, who can easily self-host and customize

[1]Code is available at https://github.com/kasnerz/factgenie/. System demonstration video: https://youtu.be/CsVcCGvOzPY.

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 13–15
September 23–27, 2024.

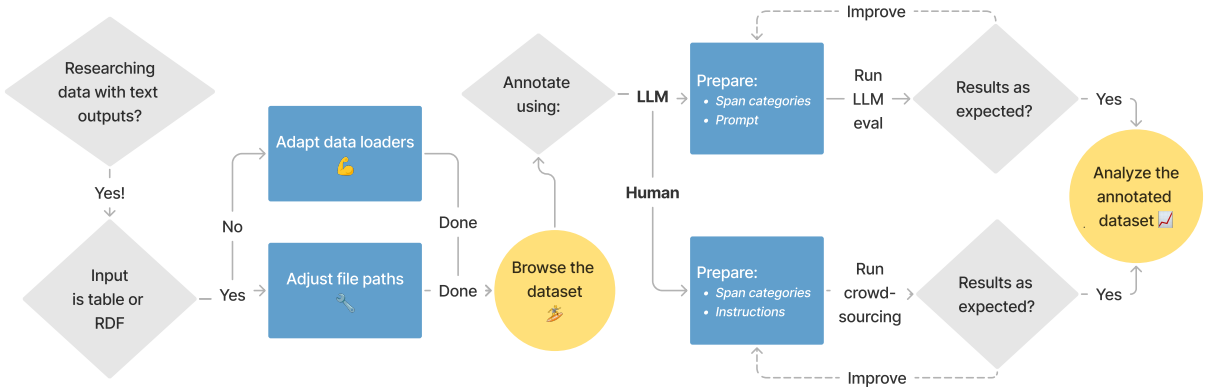


Figure 2: `factgenie` workflow. Actions needed for using `factgenie` for custom tasks are shown in blue rectangles.

it for individual experiments. The benefits of `factgenie` include:

- Visualization of input data and model outputs with a few lines of code,
- Ready-made web interface for collecting annotations from crowdsourcing services,
- Support for gathering model-based annotations from multiple LLM APIs,
- Tools for managing and visualizing collected annotations.

2 Framework

Software-wise, `factgenie` is a combination of a Flask backend and an HTML-based frontend. The frontend is powered by Boostrap 5.3 and jQuery, additionally using the YPet library for collecting span annotations. For visualizing the example input data, we use TinyHTML and Highcharts.JS.[2]

Figure 1 shows an example with weather data and the corresponding model-generated weather forecast. The model output was annotated for errors through `factgenie`. Note that the colors and labels of text span annotation categories can be customized for each set of annotations.

The framework can be used as-is or customized to cover a wide range of tasks and needs with minimal effort. To load and preview a new dataset, researchers first need to write a data loader class. Existing data loaders include various visualizations of tabular, RDF, and JSON data. As shown in Figure 2, loading a dataset in a supported format can be as easy as changing a path to the data on the file system. To add a custom dataset type, the researcher must extend the `Dataset` class. Once the dataset is loaded, `factgenie` allows data inspection and rapid prototyping of LLM annotations and crowdsourcing campaigns.

3 Human Annotations

To collect error annotation from human crowdworkers, researchers typically build custom web interfaces. With `factgenie`, researchers can easily build an annotation interface in four steps:

1. Define the campaign parameters (annotation span categories, number of examples per annotator, etc.),
2. Write instructions for the annotators,
3. Host `factgenie` on a public URL,
4. Redirect the annotators to the running `factgenie` instance.

The interface can be previewed for internal testing throughout the process. As shown in Figure 2, `factgenie` provides the feedback necessary for debugging and improving the evaluation campaign by an immediate visualization of the collected annotations.

4 LLM Annotations

It is useful to obtain annotations from LLMs in the same format as from human annotators. For that, `factgenie` provides a lightweight wrapper for model APIs.[3] The process of collecting annotations from LLMs consists of the following steps:

1. Define the campaign parameters (annotation span categories, model decoding parameters, API endpoint, etc.)

[2]In principle, `factgenie` can render datasets using any custom HTML code and JS libraries.

[3]We currently support the Ollama API for self-hosted LLMs and the OpenAI API for cloud LLMs.

2. Write the prompt and system message for the model,[4]

3. Run the LLM annotation inference.

Similarly to human annotations (Section 3), the evaluation progress can be monitored and immediately visualized.

5 Roadmap

The development of `factgenie` is ongoing and open to external developers. We are currently working on facilitating the management of evaluation campaigns by adding an option to set-up the evaluation campaign from the web interface in addition to configuration files. In the future, we plan to add more ready-made classes for data loaders, model APIs, and crowdsourcing services.

Acknowledgements

This work was funded by the European Union (ERC, NG-NLG, 101039303) and Charles University projects GAUK 40222 and SVV 260 698. It used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

References

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, and Ahmed Ali. 2024. LLMeBench: A flexible framework for accelerating LLMs benchmarking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 214–222, St. Julians, Malta.

Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations, St. Julians*, pages 150–158, Malta.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico.

Elisabeth Fittschen, Tim Fischer, Daniel Brühl, Julia Spahr, Yuliia Lysa, and Phuoc Thang Le. 2024. AnnoPlot: Interactive visualizations of text annotations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 106–114, St. Julians, Malta.

Zdeněk Kasner and Ondřej Dušek. 2024. Beyond Traditional Benchmarks: Analyzing Behaviors of Open LLMs on Data-to-Text Generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. To appear.

Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. 2024. MEGAnno+: A Human-LLM collaborative annotation system. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–176, St. Julians, Malta.

Ondrej Klejch, Eleftherios Avramidis, Aljoscha Burchardt, and Martin Popel. 2015. MT-ComparEval: Graphical evaluation interface for machine translation development. *Prague Bull. Math. Linguistics*, 104:63–74.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation, WMT 2023*, pages 768–775, Singapore.

Maxime Masson, Christian Sallaberry, Marie-Noelle Bessagnet, Annig Le Parc Lacayrelle, Philippe Roose, and Rodrigo Agerri. 2024. TextBI: An interactive dashboard for visualizing multidimensional NLP annotations in social media data. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 1–9, St. Julians, Malta.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from `https://github.com/doccano/doccano`.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 158–168, Dublin, Ireland.

František Trebuňa and Ondřej Dušek. 2023. VisuaLLM: Easy web-based visualization for neural language generation. In *Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations*, pages 6–8, Prague, Czechia.

[4]The prompt needs to instruct the model to produce JSON with a specific structure. Note that the APIs we support can ensure decoding JSON output, see, e.g., `https://platform.openai.com/docs/guides/json-mode`.

Filling Gaps in Wikipedia: Leveraging Data-to-Text Generation to Improve Encyclopedic Coverage of Underrepresented Groups

Simon Mille[1], Massimiliano Pronesti[1,2], Craig Thomson[1], Michela Lorandi[1], Sophie Fitzpatrick[3], Rudali Huidrom[1], Mohammed Sabry[1], Amy O'Riordan[3], Anya Belz[1]

[1]ADAPT, Dublin City University, [2]IBM Research, [3]Wikimedia Community Ireland

Correspondence: simon.mille@adaptcentre.ie

Abstract

Wikipedia is known to have systematic gaps in its coverage that correspond to under-resourced languages as well as underrepresented groups. This paper presents a new tool to support efforts to fill in these gaps by automatically generating draft articles and facilitating post-editing and uploading to Wikipedia. A rule-based generator and an input-constrained LLM are used to generate two alternative articles, enabling the often more fluent, but error-prone, LLM-generated article to be content-checked against the more reliable, but less fluent, rule-generated article.

1 Introduction

Knowledge equity is one of two strategic directions in Wikimedia's 2030 Movement Strategy (Wikimedia Movement, 2017). Systematic content gaps identified in the Wikimedia Knowledge Gap Taxonomy[1] relate e.g. to gender, recency, geography and language. For instance, women and non-binary people make up less than 20% of biographies on Wikipedia.[2] Addressing such gaps is important for equity and knowledge completeness, but despite increasing awareness, existing social, political and technical barriers still make it difficult to motivate and/or enable Wikipedia editors to fill them in.

Natural Language Generation (NLG) has held the ambition to help address such gaps for some time (Sauper and Barzilay, 2009; Banerjee and Mitra, 2016; Liu et al., 2018; Fan and Gardent, 2022; Shao et al., 2024), and recent work has shown it to be a valid approach (Kaffee et al., 2022). However, data-to-text NLG has a high knowledge threshold, and the more recent LLM-based NLG systems require substantial cost and energy.[3]

With the tool we report in this paper we aim to address the above issues, targeting underresourced languages and the knowledge threshold in particular. Our Wikipedia Gap Filler tool generates a draft article in Irish or English from an entity name, making it far easier for users to create texts about given entities that can be used as a starting point for a new Wikipedia page.

The contributions of this paper are: (i) the implementation and release of a tool for generating and editing text snippets on a queried entity, with a human-friendly user interface that integrates all the components of the system; (ii) the implementation and release of code for retrieving information about queried entities on DBpedia and Wikidata. The tool and source code can be found on GitHub.[4]

2 Background and Tool Overview

Kaffee et al. (2022) provided first indications that using NLG is a good strategy for filling in Wikipedia knowledge gaps. In particular they concluded that (i) Machine Translation is not adequate to create new Wikipedia pages, due to the cultural differences between the communities that speak different languages (i.e. each community has their specific points of interest; in addition, source text is not always available); (ii) providing Wikipedia editors with even just a starting sentence as in Kaffee et al.'s experiments can have a real impact on page editing; (iii) text is judged more useful than tables with raw data such as article stubs (Kaffee, 2016); and (iv) the main limitation of using NLG for creating text snippets is the lack of factual accuracy of the produced text.

The Wikipedia Gap Filler tool builds upon these ideas and goes beyond, generating the first sentence that can then be expanded upon to form a more extensive draft articles generated from a user-selected

[1] https://meta.wikimedia.org/wiki/Research:Knowledge_Gaps_Index/Taxonomy

[2] https://en.wikipedia.org/wiki/Help:Mapping_content_gaps_on_Wikimedia

[3] https://www.theguardian.com/commentisfree/article/2024/may/30/ugly-truth-ai-chatgpt-guzzling-resources-environment

[4] https://github.com/nlgcat/webnlg_demo

Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, pages 16–19
September 23–27, 2024.

set of knowledge triples. Moreover, the tool addresses issues around factual accuracy by generating two texts in parallel, one using a rule-based generator and the other using an input-constrained LLM, enabling the more fluent output from the latter to be content-checked against the more reliable output from the former. Finally, our tool provides users with a selection of entities from known gaps, encouraging them to create pages for these underrepresented entities.

Initiatives such as the WikiProject Women in Red (WiR)[5] aim to create Wikipedia content about women's biographies, women's works and women's issues. Lists of women who have no Wikipedia page, i.e. whose name on Wikipedia appears as a red link (hence the name of the initiative), were compiled by the contributors and sorted by category,[6] with 224 categories containing entities for which data is available on Wikidata. That is, for about 400,000 Women in Red, some information is available in the form of triples on (at least) Wikidata, which constitute the input to our Wikipedia Gap Filler tool. We take advantage of this resource to suggest names of WiR to users of our tool.

To use the tool, the user simply enters or selects an entity name and is then offered a list of triples (<entity, property, value>, e.g. <Barack_Obama, birthYear, 1961>) retrieved from DBpedia or Wikidata, from which those to be verbalised in the article can be selected. A wide range of entities can be queried, such as persons, places, buildings, organisations, artistic or intellectual works, fictional characters, vehicles, etc. By using only DBpedia and Wikidata contents as input, the contents of the generated texts are fully traceable, and the length of the text is customisable, by selecting more properties in the input or fewer. The user can request one or more text(s) to be generated, before editing the results in the interface, and from there, create a new Wikipedia page or enrich an existing one.

3 Interface

The Wikipedia Gap Filler app is implemented with a Python Flask back-end and a React JavaScript front-end. The user interface is intended to be very simple to use. First, to input an entity, the user has two alternatives: (i) type in an entity, or (ii) select a suggested entity from the Woman in Red frame. Then, the user can select a grammatical gender (if appropriate), an output language and an NLG system; additionally, the triple source (DBpedia Ontology, Wikipedia Infobox or Wikidata) can be specified. Changing the triple source will return different triples; if a Woman in Red is selected, the default source is Wikidata, for which we know that some triples are available (see Section 2). The user then selects some triples and clicks the Generate button to get the text(s), which can be edited. A button is also available to try and create a new Wikipedia page where the edited text can be pasted.

4 Components

In this section, we briefly describe the main components of our demo: interactive content selection, text generators, and text editing box.

4.1 Interactive content selection

Given an entity, the system first retrieves a list of properties associated with it: (i) the DBpedia/Wikidata SPARQL endpoint URL is defined, (ii) an SPARQLWrapper object is created and the query set, (iii) the query is executed, and (iv) the results are parsed and properties chosen from the DBpedia Ontology,[7] Wikidata,[8] or from a list of properties extracted from Wikipedia Infoboxes but available via DBpedia too,[9] according to the parameters defined in the user query; the Ontology tends to return less triples but of better quality than the Infobox and Wikidata sources. For the moment, the subset of queried properties is limited to about 450, roughly corresponding to the properties in the WebNLG dataset (Gardent et al., 2017; Castro Ferreira et al., 2020), on which the NLG systems we use were developed. A second query is performed on DBpedia to retrieve information about the entities of the triples (e.g. class membership, gender) which is needed by the rule-based system.[10]

4.2 Generators

Once the triples are selected, two systems can be run using the triples as input.

Rule-based pipeline. As a rule-based system, we use the FORGe multilingual generator presented in (Mille et al., 2023), which covers generation

[5] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Women_in_Red

[6] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Women_in_Red/Redlist_index

[7] http://dbpedia.org/ontology/

[8] http://www.wikidata.org/prop/direct/

[9] http://dbpedia.org/property/

[10] The *Select Grammatical Gender* field of our interface overwrites what is found in the query.

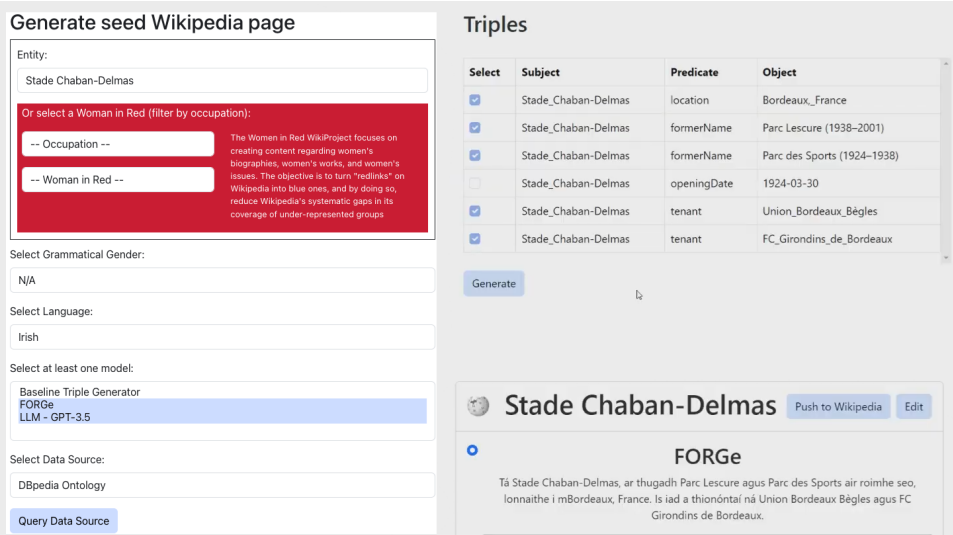


Figure 1: Screenshot of the GUI

in several languages including Irish and English. FORGe is implemented as a pipeline of components that perform subtasks such as text planning, lexicalisation, sentence structuring and surface realisation. For Irish morphology, the pipeline uses the finite-state transducers of the Irish NLP tools suite (Dhonnchadha et al., 2003). FORGe is an all-around generator that is not designed to produce specifically Wikipedia-style text, but it can be used without limitations and for free. The output text can be used as part of a seed Wikipedia page or to control the contents delivered by the LLMs.

End-to-end LLM-based generator. The current demo accesses the ChatGPT 3.5 Turbo model[11] via the aiXplain API.[12] We use the Few-Shot In-Context prompt from Lorandi and Belz (2024), which consists of a brief task description followed by two examples showing both input and output and ending with the input. We will add more models such as LLaMa2 70B chat (Touvron et al., 2023), but in general the use of LLMs will be constrained by the availability/cost of the used API.

4.3 Text editor and Wikipedia page creation

The output text is shown in a text box that has an *Edit* mode for the user to modify or combine the texts. Having logged in with their own Wikipedia editor credentials on their browser, users are able to create automatically a new empty Wikipedia page, on which they can paste the selected text.

The present tool is intended to help editors when creating new pages on Wikipedia, but it remains their responsibility to make sure that the uploaded texts respect the detailed Wikipedia edition guidelines (content, style, behaviour, etc.).[13]

5 Limitations

Some of the limitations of our tool are due to the early stage of development that it is in, but others are more general. For the first type, the tool is currently limited to English and Irish texts, and to a subset of about 14% of the DBpedia properties and 1% of the Wikidata properties. This is due to the current coverage of the rule-based generator.

The more general limitations are two-fold: (i) our tool fully depends on the availability of triples for the queried entity, but there can be very little or no information on DBpedia and Wikidata for some under-represented entities; and (ii) despite the help that NLG and LLMs can bring when creating new pages, there still are challenges and potential issues such as the introduction of societal biases and factual errors (Fan and Gardent, 2022), the attribution of the generated contents (Singh et al., 2024), or the actual impact of these technologies on Wikipedia edition (Reeves et al., 2024). Finally, the present tool is not intended to be used in an unsupervised manner, and the potential users are expected to carefully check that the contents they eventually upload to Wikipedia are correct and conform to the Wikipedia guidelines (see Section 4.3).

[11] https://platform.openai.com/docs/models/gpt-3-5-turbo

[12] https://aixplain.com/

[13] https://en.wikipedia.org/wiki/Wikipedia:List_of_guidelines

Acknowledgments

Mille's contribution was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS).

References

Siddhartha Banerjee and Prasenjit Mitra. 2016. Wikiwrite: Generating wikipedia articles automatically. In *IJCAI*, pages 2740–2746.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Elaine Uí Dhonnchadha, Caoilfhionn Nic Pháidín, and Josef Van Genabith. 2003. Design, implementation and evaluation of an inflectional morphology finite state transducer for Irish. *Machine Translation*, 18:173–193.

Angela Fan and Claire Gardent. 2022. Generating full length wikipedia biographies: The impact of gender bias on the retrieval-based generation of women biographies. *arXiv preprint arXiv:2204.05879*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Lucie Aimée Kaffee. 2016. *Generating article placeholders from Wikidata for Wikipedia: increasing access to free and open knowledge*. Ph.D. thesis, Hochschule für Technik und Wirtschaft Berlin.

Lucie-Aimée Kaffee, Pavlos Vougiouklis, and Elena Simperl. 2022. Using natural language generation to bootstrap missing wikipedia articles: A human-centric perspective. *Semantic Web*, 13(2):163–194.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Michela Lorandi and Anya Belz. 2024. High-quality data-to-text generation for severely under-resourced languages with out-of-the-box large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1451–1461, St. Julian's, Malta. Association for Computational Linguistics.

Simon Mille, Elaine Uí Dhonnchadha, Lauren Cassidy, Brian Davis, Stamatia Dasiopoulou, and Anya Belz. 2023. Generating Irish text with a flexible plug-and-play architecture. In *Proceedings of the 2nd Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 25–42, Singapore. Association for Computational Linguistics.

Neal Reeves, Wenjie Yin, Elena Simperl, and Miriam Redi. 2024. " the death of wikipedia?"–exploring the impact of chatgpt on wikipedia engagement. *arXiv preprint arXiv:2405.10205*.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216.

Yijia Shao, Yucheng Jiang, Theodore A Kanell, Peter Xu, Omar Khattab, and Monica S Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *arXiv preprint arXiv:2402.14207*.

Aakash Singh, Deepawali Sharma, Abhirup Nandy, and Vivek Kumar Singh. 2024. Towards a large sized curated and annotated corpus for discriminating between human written and ai generated texts: A case study of text sourced from wikipedia and chatgpt. *Natural Language Processing Journal*, 6:100050.

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*.

Wikimedia Movement. 2017. Wikimedia movement strategy 2017.

Author Index