INLG 2024

The 17th International
Natural Language Generation Conference

Proceedings of the Generation Challenges

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 979-8-89176-124-7

Preface

The Generation Challenges (GenChal) aim at bringing together a variety of shared-task efforts that involve the generation of natural language. This year again, the Generation Challenges was held during a special session at the 17th International Conference on Natural Language Generation (INLG 2024, September 23-27 2023). The session comprised presentations of results by the organisers of recently completed tasks, a poster session for task participants to present their submissions, as well as a presentation of a proposal for a new shared task. In 2024, we received one new shared task proposal; the proposal was reviewed positively by the four program committee members, who also provided valuable feedback to the task organisers. Three completed shared tasks are also included in these proceedings (see below), with an overview report by the organisers and participants' system descriptions (the system descriptions are included in the overview report for VGSG, see below). The system descriptions underwent a light touch review organised by each respective task organisers.

New Challenge Proposal

- Long-Form Analogy Evaluation Challenge. Bhavya Bhavya, Chris Palaguachi, Yang Zhou, Suma Bhat and ChengXiang Zhai.

Completed Challenge Overviews

- Overview of Long Story Generation Challenge (LSGC) at INLG 2024. Aleksandr Migal, Daria Seredina, Ludmila Telnina, Nikita Nazarov, Anastasia Kolmogorova and Nikolay Mikhaylovskiy.
- The 2024 GEM Shared Task on Multilingual Data-to-Text Generation and Summarization: Overview and Preliminary Results. Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Johanna Axelsson, Miruna−Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Nyunya Obonyo and Lining Zhang.
- Visually Grounded Story Generation Challenge. Xudong Hong, Khushboo Mehra, Asad Sayeed and Vera Demberg (including participants' system descriptions).

We would like to express our gratitude to the reviewers, the task organisers, as well as the INLG Programme Chairs, Workshop Chair, Publication Chair and Local Organisers for their precious during the organisation process.

Your INLG 2024 Generation Challenges chairs,
Simon Mille, Miruna−Adriana Clinciu

Organizing Committee

Generation Challenge Chairs

Simon Mille, ADAPT Research Centre, Dublin City University, Ireland
Miruna−Adriana 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, Kobayashi

Publication Chair

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

Program Committee

Anastasia Shimorina, Orange
Chris van der Lee, Tilburg University
Raquel Hervas, Complutense University of Madrid
Rudali Huidrom, Adapt Centre

Table of Contents

Long-Form Analogy Evaluation Challenge

Bhavya Bhavya[1], Chris Palaguachi[1], Yang Zhou[1], Suma Bhat[1], and ChengXiang Zhai[1]

[1]University of Illinois at Urbana-Champaign
{bhavya2, cwp5, yz96, spbhat2, czhai}@illinois.edu

Abstract

Given the practical applications of analogies, recent work has studied analogy generation to explain concepts. However, not all generated analogies are of high quality and it is unclear how to measure the quality of this new kind of generated text. To address this challenge, we propose a shared task on automatically evaluating the quality of generated analogies based on seven comprehensive criteria. For this, we will set up a leaderboard based on our dataset annotated with manual ratings along the seven criteria, and provide a baseline solution leveraging GPT-4. We hope that this task would advance the progress in development of new evaluation metrics and methods for analogy generation in natural language, particularly for education.

1 Introduction

Analogies are integral to several practical applications. In education, they help explain complex concepts by mapping them to more familiar ones (Glynn et al., 1989; Thagard, 1992) (e.g., "earth rotates on its axis like an ice skater doing a pirouette"). They also inspire creativity by connecting seemingly disparate concepts (Hey et al., 2008).

Since manually creating good analogies can be challenging and require domain expertise (Goldwater et al., 2021), recently, large language models (LLMs) like GPT-3 (Brown et al., 2020) have been used to aid with all such applications (Bhavya et al., 2022, 2023; Kim et al., 2023). They have shown great promise in generating long-form analogies (i.e., natural language analogies, typically a few paragraphs long, that describe the similarities between concepts) that are meaningful, novel (Bhavya et al., 2022, 2023) and useful for science writers (Kim et al., 2023).

However, not all automatically generated analogies are accurate or useful. Poor analogies can have negative consequences, such as, leading to misunderstanding or misconceptions (Kaufman et al., 1996). This effect can be particularly concerning when such analogies are used in educational contexts, where clarity and accuracy are crucial. Thus, evaluating the quality of generated analogies is important to identify good analogies. Although a human evaluation of all generated analogies would be ideal, it is impossible to scale up. Thus, there is a need for automatic evaluation metrics. Moreover, there is a need to develop evaluation metrics for this new type of generated text to measure the progress of analogy generation methods.

While several automatic evaluation metrics have been developed to evaluate generated text (Sai et al., 2022), they are not directly applicable to evaluate analogies. Limited work has been done on automatically evaluating generated analogies using reference-based metrics (e.g., BLEURT (Sellam et al., 2020)) and reference-free metrics (e.g., novelty estimation based on similarity to a reference corpus of analogies) (Bhavya et al., 2022, 2023). Such metrics have mostly been found to be inadequate. Moreover, it is unclear as to what precisely makes a good generated analogy since its goodness depends on multiple factors (e.g., accuracy, strength of analogical connections).

To address these challenges, we propose a new shared task for developing evaluation metrics that measure the quality of generated analogies. Specifically, we identify seven major criteria for evaluating their quality based on existing literature and our pre-pilot experiments, namely, target concept comprehensiveness, accessibility, source and target concept accuracy, mapping soundness, coherence, and repetition. Based on these evaluation criteria, we will create a dataset of manually rated analogies that are generated by models like GPT-4 in domains like science. This dataset will be used to assess the performance of automatic evaluation metrics submitted to our task.

Since LLMs have recently shown great promise in evaluating generated text (Li et al., 2024), we

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 1–16
September 23 –27, 2024.

will provide a baseline method that prompts GPT-4 for evaluation in a reference-free setting. We've found this method to be reasonably accurate based on pre-pilot experiments. But, we encourage participants to develop metrics using smaller language models and other types of models too (e.g., fact verification models for accuracy).

Similar to shared tasks on evaluation metrics for other NLG tasks (e.g., machine translation (Blain et al., 2023)), we expect our proposed task to accelerate research in both evaluation metric and text generation methods, particularly in the context of long-form analogies. More broadly, the insights from the task would also be useful for evaluating other kinds of generated long and creative text (e.g., stories). With the advent of LLMs, generation of various kinds of text has become feasible and useful for many practical applications. Therefore, we believe that this is a timely novel shared task.

2 Task Description

Given a generated analogy to explain a target concept, the overall task is to rate its quality based on defined criteria. A leader board competition would be set up to evaluate the submissions on our task and dataset. In this section, we describe the criteria we plan to use for evaluation of analogies, our datasets of human ratings and evaluation metrics to quantitatively evaluate the automatic ratings submitted to the task, and our proposed schedule.

2.1 Analogy Evaluation Criteria

Few recent work have studied evaluation of automatically generated analogies (Kim et al., 2023; Bhavya et al., 2022, 2023). Inspired by these and prior work (e.g., (Forbus and Gentner, 1989), (Glynn et al., 1989)), and further refinement based on our pre-pilot experiments (Section 3), we select seven criteria for a holistic evaluation of analogies.

Our selected criteria include measures for three main components of long-form analogies, namely, target concept, source concept, and mapping. Target is the more unfamiliar concept, and the source is the more familiar one used to explain the target. The mapping is the set of relationships or similarities between the source and the target.

For example, consider the following analogy: *"The heart is like a pump in the body's circulatory system. The pump moves fluid through a system, just as the heart moves blood through the body."* In this analogy, "the heart" is the target concept and "the pump" is the source concept. The mapping is "the pump ... the body."

We describe each of the seven criteria below. All criteria will be rated on an Ordinal scale.

Target concept comprehensiveness: Whether the analogy covers the most important details to explain the target concept.

Accessibility: Whether the analogy is familiar and easily understandable by learner.

Source Accuracy and Target Accuracy: Truthfulness of stated facts pertaining to the two analogous concepts. Instead of a single measure of overall accuracy, analyzing its two components separately is useful for applications like education, where one of them (e.g., target accuracy) is more critical.

Mapping soundness: Whether the connection between source and target is logically sound or far-fetched.

Coherence: Whether the analogy is cohesive.

Repetition: Whether the same sentence is repeated or same source concept is repeated for another target concept within the analogy.

2.2 Analogy Ratings Dataset

We plan to create an annotated dataset with human ratings to quantitatively evaluate the automatic evaluation metric submissions as described below.

Analogy Collection: To enable creation of diverse and representative data, we will include analogies that vary on the following two dimensions.

Target concept domain: Given the popularity of analogies in teaching STEM subjects(Cao et al., 2023; Glynn et al., 1989), we will focus on science and computer science domains. Depending on budget and feasibility of recruiting suitable raters, we will include other domains, such as, economics and political science. For the science domain, we will leverage existing datasets of generated analogies (Bhavya et al., 2022; Kim et al., 2023). Within a particular domain, we will consider rating analogies about target concepts of varying grade-level difficulty (e.g., beginner, intermediate, and advanced) because we expect the quality of generated analogies to differ based on them.

Generation method: Another interesting variable that impacts the quality of generated analogies is the model used for generation. For example, larger models typically generate better analogies (Bhavya et al., 2022; Kim et al., 2023). Following such work, we mainly plan to leverage the GPT-family of models, including GPT-3, GPT3.5 and GPT-4 (Brown et al., 2020; Ouyang et al., 2022; Achiam et al., 2023).

The style of generated analogy also differs based on the model and prompt used while generation. For example, GPT-3-generated analogies in one prior dataset (Bhavya et al., 2022) generally contain a single analogical comparison. While, prompts designed in another work (Kim et al., 2023) generate analogies containing several comparisons (aka "sub-analogies"). For instance, in the following analogy, *"Stratosphere is like the sky because ... Troposphere is like the earth.."*, "stratosphere" is compared to "sky", and "troposhere" to "earth". We do not plan to do an extensive exploration of prompt design, but will mostly leverage prompts from prior research.

Rating procedure: For rating analogies based on our evaluation criteria, we plan to recruit human annotators on Upwork[1], a free-lancing platform that has been used in similar prior work (Kim et al., 2023; Ouyang et al., 2022). Annotator requirements include English proficiency and prior teaching experience in the particular domain. The final set of qualified raters (up to 20 per domain) would be selected based on their performance on rating a small test batch. Each sample would be rated by three raters. We will follow other best practices for annotation and reporting (van der Lee et al., 2021; Howcroft et al., 2020), including detailed task instructions, as shown in Appendix A.1. Each rater would be paid an hourly wage of about $25-$35.

Dataset statistics: Our data would consist of validation and test sets only and no training set. To enable calibration of automatic metrics, we will use a validation set for evaluating submissions on the leader board. After the competition is over, submissions will be evaluated on a blind test set.

We plan to collect at least 1k manually rated analogies. The final number of rated analogies would mainly depend on budget and time constraints. 50% of this data would be released as the validation set and the remaining 50% would be the test set.

Evaluation of analogies would be done in a reference-free setting. This is mainly because there are many equally plausible analogies relevant for a given concept and building an exhaustive reference corpus of analogies for all concepts in the dataset is impossible. Thus, we will not release any such resources. However, participants would be free to use any external knowledge (e.g., web data).

[1]https://www.upwork.com/

2.3 Evaluation of automatic metrics

To evaluate the submitted automatic evaluation metrics, we will compare them with human ratings on each of the seven evaluation criteria using the following statistics.

Kendall's tau-b: It is commonly used to compare the rank order of automatic evaluation metrics with human ratings (Kendall, 1945; Sellam et al., 2020).

Kendall's tau-b after outlier removal: We will also measure Kendall's tau after removing outliers to avoid spurious correlations (Mathur et al., 2020).

Pairwise accuracy: To mitigate short-comings of kendall's tau in case of several ties, this metric uses pairwise accuracy, which rewards metrics for both predicting correct pair rankings and correctly predicting ties, and a tie calibration method that allows for comparing metrics that do and do not predict ties (Deutsch et al., 2023).

Krippendorff's alpha: Agreement after accounting for chance-agreements (Krippendorff, 2011).

Mean Squared Error: This measures the average difference between squared values of human and automatic ratings (James, 2013).

2.4 Baseline method

Recently, prompting LLMs like GPT-4 has shown great potential in automatically evaluating generated text based on several criteria like accuracy, coherence, and engagement in both reference-free and reference-based settings (Liu et al., 2023; Chhun et al., 2024; Li et al., 2024; Wang et al., 2023). Our pre-pilot experiments (Section 3) show reasonable results of this method on our task as well. Accordingly, we will design suitable prompts for automatic evaluation with GPT-4 based on our criteria. But, we encourage participants to leverage smaller and other kinds of models as well. For fairness, we will separately report the performances of different types of models (e.g., based on LLM size, use of external resources, etc.).

2.5 Schedule

We propose the following schedule:

September, 2024: The shared task is announced at the INLG conference. Validation data is available on the shared task website and participants can sign up for the task.

December 1st, 2024: Leaderboard based on our test sets are open for the shared task. Participants can submit their solutions and view their updated ranking on the online leaderboard based on performance on the validation set.

April 1st, 2025: Submissions are closed. Organizers conduct automatic evaluation of all submissions on the blind test set.

June 1st, 2025: Organizers will submit participant reports and overall challenge reports to INLG 2025 and present their findings.

3 Pre-pilot study

To understand the task feasibility and guide the task design, we conducted a pre-pilot study. Below, we describe the initial evaluation criteria and manual rating datasets used in this study, the results of prompting GPT-4 for automatic evaluation, and qualitative discussions to refine these criteria and finalize the ones reported in Section 2.1.

3.1 Evaluation Criteria

In addition to source and target accuracy defined in Section 2.1, we analyzed the following four criteria, guided by prior research, for the pre-pilot study.

Meaningfulness: Whether it is an accurate and coherent analogy (Bhavya et al., 2022, 2023).

Novelty: How unique is the generated text (Bhavya et al., 2023). It could be important for creative writing applications (Kim et al., 2023).

Usefulness: Overall utility of the analogy for explaining concepts, since it is one of the most important use-cases of analogies (Glynn et al., 1989).

Structural mapping consistency: It is defined by the following two constraints from Structural Mapping Engine framework (Forbus and Gentner, 1989). 1:1 constraint means that one attribute of the source concept should be connected to at most one attribute of the target and vice versa. The parallel connectivity constraint states that if two concepts are connected, then so must their attributes.

3.2 Datasets

We use the following three datasets for this study.

3.2.1 Meaningfulness and Novelty Datasets

For meaningfulness and novelty, we use datasets from previous work (Bhavya et al., 2022, 2023). In particular, one work (Bhavya et al., 2022) asked crowd-workers to rate 1608 science analogies on a binary scale for meaningfulness. Of these, 1543 are generated by GPT-3 models of various sizes (ranging from 0.3B to 175B) and 65 are human-generated ones scraped from online websites like `chegg.com`. We call this dataset as **BAM** for Binary Analogy Meaningfulness.

In another work (Bhavya et al., 2022), crowd-workers were asked to rate 347 GPT-3-generated science analogies on both meaningfulness and novelty on a scale of 1-4. We call this dataset as **OAMN** for Ordinal Analogy Meaningfulness and Novelty. Three annotators rated each analogy in both cases.

Table 1: Krippendorff's alpha (α) between human annotator (ann.) and GPT-4 on automatically and human generated analogies in BAM.

	Auto-generated	Human-generated
All ann.	0.49	0.22
GPT-4 v. ann.	0.56 ± 0.009	0.35 ± 0.045

Table 2: Krippendorff's alpha (α) and Kendall's tau (τ) between human annotator (ann.) and GPT-4 on OANM.

	Meaningfulness		Novelty	
	α	τ	α	τ
All ann.	0.247	-	0.4	-
GPT-4 v. ann.	0.46 ± 0.02	0.48 ± 0.02	0.33 ± 0.003	0.33 ± 0.001

3.2.2 Multi-Aspect Analogy Annotation for Education (ManaEd)

For the remaining four criteria, we manually rate a 50 analogies about 7 target concepts released by another work (Kim et al., 2023).[2] Two researchers, a graduate student in Educational Psychology and an undergraduate in Computer Science, rate each analogy on a scale of 1-4 for all criteria. Source and target accuracy were rated at the sub-analogy level (refer Section 2.2, Generation method).

3.3 Experiments

Using the above datasets, we study the feasibility of prompting GPT-4 for automatic analogy evaluation, and the suitability of our evaluation criteria based on the quantitative and qualitative results.

Methodology: We leverage prompt templates from recent work on prompting GPT-4 for text evaluation (Liu et al., 2023), and conduct light prompt-tuning, including the use of suitable instructions and examples for our task. The best performing prompts for each criteria are shown in Appendix A.2.

We quantitatively compare GPT-4 (gpt4-0125-preview) ratings with average human ratings based on Krippendorff's alpha and Kendall's tau. As an upper limit, we also report the inter-annotator agreements and correlations (if applicable). Further, qualitative discussions and analysis of manual

[2] Although they release manual ratings by science writers on some criteria, those are not usable because ratings cannot be mapped to their corresponding analogies.

Table 3: Krippendorff's alpha (α) and Kendall's tau (τ) between human annotators (ann.) and GPT-4 on MANAED

	Structural Consistency		Usefulness		Source Accuracy		Target Accuracy	
	α	τ	α	τ	α	τ	α	τ
All ann.	0.6	58	0.62	0.56	0.51	0.48	0.49	0.48
GPT-4 v. ann.	0.23 ± 0.05	0.2 ± 0.05	0.29 ± 0.07	0.25 ± 0.06	0.37 ± 0.001	0.33 ± 0.001	0.31 ± 0.01	0.3 ± 0.02

ratings were conducted to refine criteria.

Results: From Tables 1, 2 and 3, on all the six criteria, GPT-4 generally achieves fair to moderate agreements and correlations (Landis and Koch, 1977; Schober et al., 2018), suggesting its feasibility to use as a baseline method.

On meaningfulness, from Tables 1 and 2, we observe that GPT-4's agreement and correlation with human ratings is comparable to that among humans. Due to this already strong performance of GPT-4, we discard this criteria for the main task.

Results for novelty and other remaining criteria are in Tables 2 and 3, respectively. For these criteria, there is a gap between GPT-4 and human performance, suggesting room for research.

After discussions, we discard novelty because it depends on training and reference dataset. For instance, an analogy can be considered not novel (or novel) depending on whether the model that generates it has seen it during training (or not).

Further, by analyzing annotator disagreements, we identified usefulness to be highly subjective because it spans multiple aspects. So, we identify the following three salient aspects, aligned with prior research (Glynn et al., 1989), that impact utility of long-form analogies for education, in addition to our other included criteria: "target comprehensiveness", "accessibility", and "mapping soundness".

Additionally, the two structural mapping constraints are decoupled and adapted for LLM-generated analogies. In this way, we finalize "repetition", corresponding to 1:1 constraint, and "coherence", corresponding to parallel connectivity.

4 Related Work

Prior work has studied the modeling and generation of various forms of analogies (Mitchell, 2021), such as, analogies between structured representations of concepts (Forbus et al., 2017), relational and proportional analogies (e.g., king:queen::man:woman) (Ushio et al., 2021; Yuan et al., 2023; Chen et al., 2022), analogies relating longer text, such as, two sentences or stories (Jiayang et al., 2023; Wijesiriwardene et al., 2023; Sultan et al., 2024), and more recently, *long-form* analogies that explain the relation between concepts using natural language (Seals and Shalin, 2023; Bhavya et al., 2022, 2023; Kim et al., 2023; Cao et al., 2023). We aim to evaluate long-form analogies that are typically a few paragraphs long.

Human evaluation of generated text, although ideal, is highly resource extensive. Accordingly, several automatic metrics have been developed for evaluating generated text (Sai et al., 2022), and shared tasks have been established to drive such efforts (Blain et al., 2023). We build upon recent work on holistic evaluation of other types of figurative and creative text (Chhun et al., 2022; He et al., 2023), because it enables a fine-grained evaluation. However, for automatic evaluation of generated long-form analogies, there has been very limited work (Bhavya et al., 2022, 2023; Kim et al., 2023). We compile and refine seven major evaluation criteria based on these and prior work on analogical modeling and reasoning (Falkenhainer et al., 1989; Glynn et al., 1989), aim to extend their datasets both in the number of samples and ratings based on our criteria, and call for development of suitable automatic evaluation metrics.

5 Conclusion

We propose a new shared task for development of automatic metrics to evaluate generated long-form analogies, which describe the analogical relation between concepts in natural language, on seven comprehensive criteria. The submissions would be evaluated based on their agreement with human ratings on our datasets. With this shared task, we hope to accelerate the progress in evaluation metrics and generation methods for long-form analogies.

6 Acknowledgment

This material is based upon work supported by the National Science Foundation and the Institute of Education Sciences under Grant #2229612. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation or the U.S. Department of Education.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bhavya Bhavya, Jinjun Xiong, and Chengxiang Zhai. 2022. Analogy generation by prompting large language models: A case study of instructgpt. *arXiv preprint arXiv:2210.04186*.

Bhavya Bhavya, Jinjun Xiong, and Chengxiang Zhai. 2023. Cam: A large language model-based creative analogy mining framework. In *Proceedings of the ACM Web Conference 2023*, pages 3903–3914.

Frederic Blain, Chrysoula Zerva, Ricardo Rei, Nuno M Guerreiro, Diptesh Kanojia, José GC de Souza, Beatriz Silva, Tânia Vaz, Yan Jingxuan, Fatemeh Azadi, et al. 2023. Findings of the wmt 2023 shared task on quality estimation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 629–653.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen Cao, Zijian Ding, Gyeong-Geon Lee, Jiajun Jiao, Jionghao Lin, and Xiaoming Zhai. 2023. Elucidating stem concepts through generative ai: A multi-modal exploration of analogical reasoning. *arXiv preprint arXiv:2308.10454*.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-kar: A benchmark for rationalizing natural language analogical reasoning. *arXiv preprint arXiv:2203.08480*.

Cyril Chhun, Pierre Colombo, Chloé Clavel, and Fabian M Suchanek. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. *arXiv preprint arXiv:2208.11646*.

Cyril Chhun, Fabian M Suchanek, and Chloé Clavel. 2024. Do language models enjoy their own stories? prompting large language models for automatic story evaluation. *arXiv preprint arXiv:2405.13769*.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Brian Falkenhainer, Kenneth D Forbus, and Dedre Gentner. 1989. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63.

Kenneth D Forbus, Ronald W Ferguson, Andrew Lovett, and Dedre Gentner. 2017. Extending sme to handle large-scale cognitive modeling. *Cognitive Science*, 41(5):1152–1201.

Kenneth D Forbus and Dedre Gentner. 1989. Structural evaluation of analogies: What counts. In *Proceedings of the eleventh annual Conference of the Cognitive Science Society*, volume 34, pages 341–348.

Shawn M Glynn, Bruce K Britton, Margaret Semrud-Clikeman, and K Denise Muth. 1989. Analogical reasoning and problem solving in science textbooks. *Handbook of creativity*, pages 383–398.

Micah B Goldwater, Dedre Gentner, Nicole D LaDue, and Julie C Libarkin. 2021. Analogy generation in science experts and novices. *Cognitive Science*, 45(9):e13036.

Qianyu He, Yikai Zhang, Jiaqing Liang, Yuncheng Huang, Yanghua Xiao, and Yunwen Chen. 2023. Hauser: Towards holistic and automatic evaluation of simile generation. *arXiv preprint arXiv:2306.07554*.

Jonathan Hey, Julie Linsey, Alice M Agogino, and Kristin L Wood. 2008. Analogies and metaphors in creative design. *International Journal of Engineering Education*, 24(2):283.

David M Howcroft, Anya Belz, Miruna Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel Van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *13th International Conference on Natural Language Generation 2020*, pages 169–182. Association for Computational Linguistics.

G James. 2013. An introduction to statistical learning.

Cheng Jiayang, Lin Qiu, Tsz Ho Chan, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, et al. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. *arXiv preprint arXiv:2310.12874*.

David R Kaufman, Vimla L Patel, and Sheldon A Magder. 1996. The explanatory role of spontaneously generated analogies in reasoning about physiological concepts. *International Journal of Science Education*, 18(3):369–386.

Maurice G Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Jeongyeon Kim, Sangho Suh, Lydia B Chilton, and Haijun Xia. 2023. Metaphorian: Leveraging large language models to support extended metaphor creation for science writing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pages 115–135.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

J Richard Landis and Gary G Koch. 1977. An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39.

Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768.

SM Seals and Valerie L Shalin. 2023. Long-form analogies generated by chatgpt lack human-like psycholinguistic properties. *arXiv preprint arXiv:2306.04537*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Oren Sultan, Yonatan Bitton, Ron Yosef, and Dafna Shahaf. 2024. Parallelparc: A scalable pipeline for generating natural-language analogies. *arXiv preprint arXiv:2403.01139*.

Paul Thagard. 1992. Analogy, explanation, and education. *Journal of Research in science Teaching*, 29(6):537–544.

Asahi Ushio, Luis Espinosa-Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? *arXiv preprint arXiv:2105.04949*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Thilini Wijesiriwardene, Ruwan Wickramarachchi, Bimal Gajera, Shreeyash Gowaikar, Chandan Gupta, Aman Chadha, Aishwarya Naresh Reganti, Amit Sheth, and Amitava Das. 2023. Analogical-a novel benchmark for long text analogy evaluation in large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3534–3549.

Siyu Yuan, Jiangjie Chen, Changzhi Sun, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2023. Analogykb: Unlocking analogical reasoning of language models with a million-scale knowledge base. *arXiv preprint arXiv:2305.05994*.

A Appendix

A.1 Sample instructions for manually rating analogies

Task Overview:

By connecting abstract or unfamiliar concepts (called the target) to more familiar ones (called the source), analogies play a huge role in education as they help with understanding concepts, problem-solving, increasing learners' interest and motivation.

For example, "The heart is like a pump in the body's circulatory system.The pump moves fluid through a system, just as the heart moves blood through the body."

In this analogy, the heart is the target concept and the pump is the source concept. The mapping is the set of relationships or correspondences between the source and the target. In the example above, the mapping is: The pump moves fluid through a system, just as the heart moves blood through the body.

Your task is to rate analogies based on seven criteria defined below.

Target concept comprehensiveness/scope: Whether the analogy covers the most important details to explain the target concept

1 - Does not cover anything; not suitable for anyone

2 - Covers sufficient details for elementary school

students and beginners
3 - Covers sufficient details for middle school students and intermediate learners
4 - Covers sufficient details for high school students and advanced learners
Examples:
1- Does not cover anything; not suitable for anyone:
Target concept: Photosynthesis, Analogy: "Photosynthesis is like a tree eating sunshine."
This analogy is too simplistic and doesn't cover any important details about photosynthesis. It doesn't explain the process, components involved, or the purpose of photosynthesis.
2 - Covers sufficient details for elementary school students and beginners:
Target concept: The water cycle, Analogy: "The water cycle is like a never-ending merry-go-round. Water from puddles, lakes, and oceans gets warmed by the sun and turns into vapor that rises into the sky. It forms clouds, and when the clouds get heavy, the water falls back to Earth as rain or snow, starting the ride all over again." This analogy covers basic components of the water cycle (evaporation, condensation, precipitation).
3- Covers sufficient details for middle school students and intermediate learners:
Target concept: The immune system, Analogy: "The immune system is like a well-organized army protecting a country. It has scouts (white blood cells) that patrol the body looking for invaders (pathogens). When they spot an enemy, they alert the command center (lymph nodes) which then sends out specialized troops (antibodies) to fight the specific invader. The army also keeps records of past battles (memory cells) to respond more quickly if the same invader returns."
This analogy covers more complex aspects of the immune system, including different types of cells and their functions, making it suitable for intermediate learners.
4- Covers sufficient details for high school students and advanced learners:
Target concept: DNA replication, Analogy: "DNA replication is like a highly efficient book-copying process in a specialized library. The original DNA double helix is the master book, which is carefully unzipped (by helicase enzymes) into two single strands. Each strand serves as a template for creating a new complementary strand. Skilled workers (DNA polymerase) move along each template, reading the sequence and adding corresponding nucleotides to build the new strands. They work in a specific direction (5' to 3'), creating a continuous leading strand and a fragmented lagging strand (Okazaki fragments). Proofreaders (exonuclease function) check for errors, and librarians (ligase enzymes) connect the fragments. The result is two identical copies of the original DNA book, each containing one old and one new strand."
This analogy covers detailed aspects of DNA replication, including enzyme names, directionality, and specific processes like the formation of Okazaki fragments. It's suitable for advanced learners or high school students studying biology.
Accessibility:
Whether the analogy is familiar and easily understandable by learner
1 - Easily understandable by elementary school students and beginners
2 - Easily understandable by middle school students and intermediate learners
3 - Easily understandable by high school students and advanced learners
Examples
1 - (Elementary school/Beginners):
Target concept: The water cycle, Analogy: "The water cycle is like a merry-go-round. Water goes up into the sky, forms clouds, falls as rain, and then goes back up again, just like how you go up and down on a merry-go-round."
This analogy uses a merry-go-round, which is a simple, familiar concept for young children.
2 - (Middle school/Intermediate):
Target concept: Photosynthesis, Analogy: "Photosynthesis is like a plant's kitchen. The leaves are the chef, sunlight is the stove, water and carbon dioxide are the ingredients, and glucose is the meal the plant makes for itself."
This analogy uses the concept of a kitchen, which is familiar to most people but requires a slightly more abstract understanding than the merry-go-round example. It introduces more specific terms (like "ingredients" and "glucose") and requires understanding the idea of transforming ingredients into a meal.
3 - (High school/Advanced):
Target concept: DNA replication, Analogy: "DNA replication is like creating a backup of an important computer file. The original DNA strand serves as a template, much like the original file, while enzymes act as the copying software, creating an exact duplicate to ensure the genetic information is

preserved and can be passed on."
This analogy uses the concept of computer file backup, which is more technologically advanced and less universally familiar than the previous examples.

Mapping soundness:

Whether the similarity between source and target is logically sound

1- No mapping; connections are illogical or nonexistent. Includes cases where the text is an example, definition, or a tautology.

2 - Weak mapping; similarities are far-fetched or barely logical

3 - Moderate mapping; similarities are mostly logical with minor logical stretches

4 - Strong mapping; similarity between source and target concept is logically sound

Examples

1- No mapping; connections are illogical or nonexistent:

Analogy: "Life is like a banana because both have letters."

This is illogical as the presence of letters in words is not a meaningful connection between life and bananas.

2- Weak mapping; similarities are far-fetched or barely logical:

Analogy: "A computer is like a sandwich because both can be found on a desk."

While it's possible to find both on a desk, this similarity is superficial and doesn't relate to their core functions or properties.

3 - Moderate mapping; similarities are mostly logical with minor logical stretches:

Analogy: "A library is like a beehive because both are organized systems where individuals work together to store and access valuable resources."

This analogy has logical similarities in terms of organization and collective effort, but stretches the comparison between books/information and honey/pollen.

4- Strong mapping; similarity between source and target concept is logically sound:

Analogy: "The human brain is like a computer because both process information, store memories, and can be overloaded with too much input."

This analogy draws clear, logical parallels between the functions of a brain and a computer, with sound similarities in their core operations.

Coherence:

Whether the whole analogy is cohesive

1 - All the sentences or sub-analogies are disconnected

2 - Most of the sentences or sub-analogies are disconnected

3 - Some of the sentences or sub-analogies are disconnected

4 - None of the sentences or sub-analogies are disconnected

Examples

1 - All the sentences or sub-analogies are disconnected:

Target concept: Learning, Analogy: "Learning is like climbing a mountain. Fish swim in schools. Knowledge weighs nothing. Time is money."

2 - Most of the sentences or sub-analogies are disconnected:

Target concept: Career growth, Analogy: "Career growth is like tending a garden. You need to plant seeds of opportunity. Success doesn't happen overnight. A rolling stone gathers no moss."

3 - Some of the sentences or sub-analogies are disconnected:

Target concept: Problem-solving, Analogy: "Problem-solving is like untangling a knot. You need patience and persistence to work through the complications. Sometimes you need to approach it from a different angle. Every cloud has a silver lining."

4 - None of the sentences or sub-analogies are disconnected:

Target concept: The internet, Analogy: "The internet is like a vast ocean of information. Websites are islands, each with their own unique landscape and inhabitants. Search engines are the ships that navigate these waters, helping users chart a course to their desired destination. Social media platforms are bustling ports where people from all over this digital world gather to exchange ideas and experiences."

Repetition:

Whether the same sentence is repeated or same source concept is repeated for another target concept within the analogy

1 - All the sentences or source concepts are repeated

2 - Most of the sentences or source concepts are repeated

3 - Some of the sentences or source concepts are repeated

4 - None of the sentences or source concepts are repeated

Examples

1 - All the sentences or source concepts are

repeated:
Target: The Atom, Analogy: "The atom is like the solar system. The nucleus is like the solar system. Electrons are like the solar system. Protons are like the solar system. Neutrons are like the solar system."
2 - Most of the sentences or source concepts are repeated:
Target: The Human Body, Analogy: "The human body is like a machine. The brain is like a machine. The heart is like a pump. The lungs are like bellows. The digestive system is like a machine."
3 - Some of the sentences or source concepts are repeated:
Target: The Solar System Analogy: "The Solar System is like a family. The Sun is like a parent. Planets are like children. Moons are like children. Asteroids are like extended family members. Comets are like distant relatives."
4 - None of the sentences or source concepts are repeated:
Target: Cell Structure, Analogy: "A cell is like a city. The nucleus is like the city hall containing DNA blueprints. Mitochondria are like power plants generating energy. The cell membrane is like the city walls controlling what enters and exits. Ribosomes are like factories producing proteins."

Target Accuracy:

Truthfulness of all facts pertaining to target concept.
N/A - Target missing
1 - None of the facts stated about the target are accurate
2 - Some of the facts stated about the target are accurate
3 - Most of the facts stated about the target are accurate
4 - All of the facts stated about the target are accurate

Examples

N/A - Target missing:
Target: Photosynthesis, Analogy: "A refrigerator keeps food cold to prevent spoilage."
Analogy is not about photosynthesis
1 - None of the facts stated about the target are accurate: Target: Photosynthesis, Analogy: "Photosynthesis is like a furnace burning wood to generate heat and ash."
This analogy is completely inaccurate about the energy conversion and processes involved in photosynthesis.
2 - Some of the facts stated about the target are accurate:
Target: Photosynthesis, Analogy: "Photosynthesis is like a factory where plants produce packaged goods by absorbing water and heat from the soil."
Plants produce energy, not packaged goods. While plants do absorb water and use energy, the source of energy is sunlight, not heat from the soil.
3 - Most of the facts stated about the target are accurate:
Target: Photosynthesis, Analogy: "Photosynthesis is like a solar-powered factory. The leaves act as solar panels, capturing sunlight energy. The process occurs in special organelles called mitochondria, and the green pigment responsible for absorbing light is called chlorophyll."
There is one significant inaccuracy: the process occurs in chloroplasts, not mitochondria.
4 - All of the facts stated about the target are accurate:
Target: Photosynthesis, Analogy: "Photosynthesis is like a solar-powered factory. Plants use sunlight energy to convert carbon dioxide and water into glucose and oxygen. This process takes place in chloroplasts, where the green pigment chlorophyll absorbs sunlight to drive the chemical reactions."
This analogy accurately describes the inputs, outputs, energy source, and location of the photosynthesis process.

Source Accuracy:

Truthfulness of all facts pertaining to source concept.
N/A - Source missing
1 - None of the facts stated about the source are accurate
2 - Some of the facts stated about the source are accurate
3 - Most of the facts stated about the source are accurate
4 - All of the facts stated about the source are accurate

Examples

N/A - Source missing:
Target: Lightning, Analogy: "Lightning is like a big spark."
Lighting is an example of a big spark, they are not different concepts.
1 - None of the facts stated about the source are accurate:
"The solar system is like a beehive, where the queen bee (the Sun) stays stationary in the center while worker bees (planets) fly in concentric circular paths around her at the same speed."

This analogy contains no accurate facts about beehives. Queen bees don't stay stationary in the center, worker bees don't fly in concentric circles around the queen, and they certainly don't all move at the same speed.
2 - Some of the facts stated about the source are accurate:
Target: Solar system, Analogy: "The solar system is like a classroom, where the teacher (the Sun) stands at the front, and students (planets) sit in rows, getting colder as they sit further back. Each student spins in their chair while moving around the classroom."
Some facts are accurate: teachers often stand at the front, and students do sit in rows. However, students don't typically spin in their chairs or move around the classroom, and the temperature doesn't necessarily decrease as you move further back.
3 - Most of the facts stated about the source are accurate:
Target: Solar system, Analogy: "The solar system is like a playground merry-go-round, where the center pole (the Sun) remains fixed while children (planets) spin around it. The kids closer to the center (inner planets) complete their revolutions faster than those at the edge (outer planets). Some children have backpacks (moons) attached to them."
All facts about the merry-go-round are correct except that kids closer to the center do not complete revolutions faster. All riders complete one revolution in the same amount of time, regardless of their position.
4 - All of the facts stated about the source are accurate:
Target: Solar system, Analogy: "The solar system is like a clock, with the central point (Sun) remaining stationary while the hands (planets) move around it at different speeds. Each hand (planet) follows a predictable path, completing full revolutions in varying amounts of time."
This analogy uses entirely accurate facts about the clock.

A.2 Prompt Templates for Pre-pilot Study

You will be given one piece of text written to explain a target concept.

Your task is to rate the text on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Meaningful analogy (1 or 0) - Whether the given text is a meaningful analogy or not. Some examples of text that is not a meaningful analogy include the following cases:
The text is not actually an analogy. It could be a definition, example, tautology, etc.
The text contains little to no relevant information pertaining to the target concept.
Important details about the analogous concepts are either incorrect or missing, or the provided explanation was insufficient, making the analogy completely wrong or weak at best.
The text is completely incoherent or gramatically incorrect.

Evaluation Steps:

1. Read the given text carefully.
2. Assign a 0 or 1 score for the meaningful analogy criteria.

Examples:

Text: Cytoplasm is like a school secretary with the difference that cytoplasm is in a liquid form and school secretary is in a dry form.

Evaluation Form:

- Meaningful analogy: 0

Text: Macrophages are similar to guards in that they are both responsible for protecting the body from harm. Macrophages are the first line of defense against infection, while guards are responsible for protecting people and property.

Evaluation Form:

- Meaningful analogy: 1

=========================
Target: '{{Target}}'

Text:

{{Document}}

Evaluation Form:

- Meaningful analogy:

Figure 1: Prompt template used for BAM

You will be given one piece of text written to explain a target concept.

Your task is to rate the text on two metrics.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Meaningful analogy (1-4) - Whether the given text is a meaningful (i.e., valid and correct) analogy, where,
1 means Strongly Disagree that text contains meaningful analogy,
2 means Somewhat Disagree that text contains meaningful analogy,
3 means Somewhat Agree that text contains meaningful analogy,
4 means Strongly Agree that text contains meaningful analogy.
Some examples of text that is not a meaningful analogy include the following cases:
The text is not actually an analogy. It could be a definition, example, tautology, etc.
The text contains little to no relevant information pertaining to the target concept.
Important details about the analogous concepts are either incorrect or missing, or the provided explanation was insufficient, making the analogy completely wrong or weak at best.
The text is completely incoherent or gramatically incorrect.

Novelty (1-4) - How novel is the text, i.e., can similar text be found online?
1 means the same text (potentially paraphrased) is found on the web,
2 means similar text is found on the web,
3 means no similar text is found online but text is straightforward to infer from the content found online,
4 means no remotely similar text is found online and text is not straightforward to infer from the content found online.

Evaluation Steps:

1. Read the given text carefully.
2. Assign a score on a scale of 1-4 for the meaningful analogy criteria.
3. Assign a score on a scale of 1-4 for the novelty criteria.

Examples:

Text: DNA replication can be thought of as a photocopier. The DNA molecule is like the original document, and each strand of DNA is like one copy of the document. During replication, the two strands are separated, and new copies of each strand are created.

Evaluation Form:

- Meaningful analogy: 4
- Novelty: 1

Text: Breathing mechanism of frogs can be analogy to bellows of blacksmith. Just like bellows, the frog's lungs are inflated and deflated by muscles that run along either side of its ribcage. When the frog inhales, the muscles contract, pushing air into the lungs. When it exhales, the muscles relax and air is forced out.

Evaluation Form:

- Meaningful analogy: 4
- Novelty: 4

Text: In computing, an operating system kernel is the core of a computer operating system. It is responsible for managing hardware and software resources and providing common services for application programs. The kernel performs its tasks in cooperation with device drivers, which are modules that load into the kernel to provide specific functions, such as access to the disk drive or network card.

Evaluation Form:

- Meaningful analogy: 1
- Novelty: 1

========================
Target: '{{Target}}'

Text:

{{Document}}

Evaluation Form:

Figure 2: Prompt template used for OAMN

You will be given one analogy written to explain a target concept.

Your task is to rate the analogy on four metrics.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Target Accuracy (1-4) - The accuracy of facts about the target concept. Penalize factually incorrect text about the target concept.

Source Accuracy (1-4) - The accuracy of facts about the source concept. Penalize factually incorrect text about the source concept. If a separate source concept is not found (e.g., source concept is missing or the target concept is compared to itself), set this score to -1.

Mapping Consistency (1-4) - Structural consistency of the mapping between source and target concepts. Penalize if the source concepts of the sub-analogies are disconneted (i.e., do not coherently consitute a single concept). Also, penalize if 1:1 mapping is not found in the sub-analogies (i.e., if the same source or target concept is used in multiple sub-analogies).

Usefulness (1-4) - The usefulness of the analogy for explaining the concept.

Evaluation Steps:

1. Read the analogy carefully and identify all the sub-analogies.
2. Read each sub-analogy and identify the target and source concept (the concept being compared to the target).
3. For each sub-analogy, write it and assign a score for its target accuracy on a scale of 1 to 4, where 1 is the lowest and 4 is the highest based on the Evaluation Criteria.
4. For each sub-analogy, write it and assign a score for its source accuracy on a scale of 1 to 4, where 1 is the lowest and 4 is the highest, or set it to -1 based on the Evaluation Criteria .
5. Assign a score for the overall mapping consistency on a scale of 1 to 4, where 1 is the lowest and 4 is the highest as per the Evaluation Criteria.
6. Assign a score for the overall usefulness on a scale of 1 to 4, where 1 is the lowest and 4 is the highest as per the Evaluation Criteria.

Example:

Analogy Text:
The atmosphere is like a hug because it is warm and comforting. The thermosphere is like the top of a mountain because it is the highest point. The mesosphere is like the middle of a journey because it is the middle point. The troposphere is like the bottom of the ocean because it is the lowest point.

Evaluation Form:

- Sub-analogy 1: The atmosphere is like a hug because it is warm and comforting.
  - Source Accuracy: 4
  - Target Accuracy: 2
- Sub-analogy 2: The thermosphere is like the top of a mountain because it is the highest point.
  - Source Accuracy: 4
  - Target Accuracy: 1
- Sub-analogy 3: The mesosphere is like the middle of a journey because it is the middle point.
  - Source Accuracy: 4
  - Target Accuracy: 4
- Sub-analogy 4: The troposphere is like the bottom of the ocean because it is the lowest point.
  - Source Accuracy: 4
  - Target Accuracy: 4
- Mapping Consistency: 2
- Usefulness: 3

========================

Target: '{{Target}}'

Analogy Text:

{{Document}}

Evaluation Form:

- Sub-analogy 1:

Figure 3: Best performing prompt template for structural consistency on MANAED

You will be given one analogy written to explain a target concept.

Your task is to rate the analogy on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Usefulness (1-4) - The usefulness of the analogy for explaining the concept.

Evaluation Steps:

1. Read the analogy carefully.
2. Assign a score for the overall usefulness on a scale of 1 to 4, where 1 is the lowest and 4 is the highest as per the Evaluation Criteria.

Example:

Analogy Text:
The atmosphere is like a hug because it is warm and comforting. The thermosphere is like the top of a mountain because it is the highest point. The mesosphere is like the middle of a journey because it is the middle point. The troposphere is like the bottom of the ocean because it is the lowest point.

Evaluation Form:

- Usefulness: 3

========================

Analogy Text:

{{Document}}

Evaluation Form:

- Usefulness:

Figure 4: Best performing prompt template for usefulness on MANAED

You will be given one analogy written to explain a target concept.

Your task is to rate the analogy on four metrics.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Source Accuracy {-1, 1-4} - The accuracy of facts about the source concept. Penalize factually incorrect text about the source concept. If a separate source concept is not found (e.g., source concept is missing or the target concept is compared to itself), set this score to -1.

Target Accuracy (1-4) - The accuracy of facts about the target concept. Penalize factually incorrect text about the target concept.

Evaluation Steps:

1. Read the analogy carefully.
2. Identify all facts related to the source concept (the concept being compared to the target).
3. Assign a score for its source accuracy on a scale of 1 to 4, where 1 is the lowest and 4 is the highest, or set it to -1 based on the Evaluation Criteria.
4. Read each sub-analogy and identify all facts related to the target concept.
5. Assign a score for the target accuracy on a scale of 1 to 4, where 1 is the lowest and 4 is the highest.

Examples:

Analogy Text: The atmosphere is like a blanket because it surrounds and protects us.

Evaluation Form:
- Source Accuracy (blanket): 4
- Target Accuracy (atmosphere): 4

Analogy Text: System software is like the sugar for a cake because it helps to sweeten the final product.

Evaluation Form:
- Source Accuracy (sugar): 4
- Target Accuracy (system software): 1

Analogy Text: The moons are the cousins because they orbit the planets and are much smaller than the planets.

Evaluation Form:
- Source Accuracy (cousins): 1
- Target Accuracy (moons): 4

========================

Target: '{{Target}}'

Analogy Text:

{{Document}}

Evaluation Form:

Figure 5: Best performing prompt template for source and target accuracy on MANAED

The 2024 GEM Shared Task on Multilingual Data-to-Text Generation and Summarization: Overview and Preliminary Results

Simon Mille[1], João Sedoc[2], Yixin Liu[3], Elizabeth Clark[4], Agnes Axelsson[5], Miruna Clinciu[6], Yufang Hou[7], Saad Mahamood[8], Ishmael Obonyo[9], Lining Zhang[2]

[1]ADAPT, Dublin City University, [2]New York University, [3]Yale University, [4]Google DeepMind, [5]Delft University of Technology, [6]Heriot Watt University, [7]IBM Research, [8]Trivago, [9]Technical University of Kenya

Correspondence: simon.mille@adaptcentre.ie, jsedoc@stern.nyu.edu

Abstract

We present an overview of the GEM 2024 shared task, which comprised both data-to-text generation and text summarization. New datasets were compiled specifically for the task to reduce the data contamination issue in large language models (LLMs) that the participants were likely to use. The paper describes the tasks, datasets, participating systems, evaluation methods, and some preliminary results. The full results will be presented at INLG '24. In this paper, we provide (i) the metrics results for English texts on six different data-to-text test sets for which we collected new reference texts, and (ii) the metrics results for Swahili on the text summarization test set.

1 Introduction

Since its inception, the Generation, its Evaluation and Metrics initiative (GEM (Gehrmann et al., 2021)) has had the objective to contribute to measuring progress in the field of Natural Language Generation (NLG), via the creation of datasets and tools for automatic and human assessments of text generation systems on different NLG tasks (McMillan-Major et al., 2021; Mille et al., 2021; Dhole et al., 2023; Gehrmann et al., 2022, 2023; Zhang et al., 2023; Nawrath et al., 2024). In the past few years, large language models (LLMs) have been widely used in NLG; they have been trained on enormous amounts of data, to the point that it can be unclear what they have seen or not during training time (Balloccu et al., 2024). To challenge these models, the NLG community has recently been developing methods for creating ad-hoc input data that the models cannot have been exposed to. For instance, Axelsson and Skantze (2023) propose to build dynamically counterfactual and fictional inputs for data-to-text generation, and Kasner and Dušek (2024) released a tool for collecting new test sets using public APIs; the creation or compilation of reference texts for the collected inputs remains an open issue.

In parallel, the interest for multilingual Natural Language Processing has been growing, with the organisation of shared tasks that included under-resourced languages, such as Universal Dependency parsing (Zeman et al., 2018) for syntactic parsing, MSR (Mille et al., 2018) for surface realisation, LowResourceEval (Klyachko et al., 2020) for morphological analysis, LowresMT (Ojha et al., 2020, 2021) and WMT (Libovický and Fraser, 2021) for machine translation, as well as WebNLG (Cripwell et al., 2023) for data-to-text generation.

Inspired by the current state of affairs, this edition of the GEM shared task[1] has two main objectives: (i) to assess LLMs—and more broadly NLG systems—using new ad-hoc datasets that no model could have already been exposed to, and (ii) to encourage participants to come up with approaches suitable across languages (including low-resource languages). We created data for two tasks, namely *data-to-text generation* and *text summarization*. The data-to-text task comprises 6 types of inputs: in-domain factual data, in-domain counterfactual data, in-domain fictional data, out-of-domain factual data, out-of-domain counterfactual data, and out-of-domain fictional data. We accepted output texts in 9 languages: Arabic, English, Chinese, German, Hindi, Korean, Russian, Spanish and Swahili; small sets of new human-written references were compiled for all 6 test sets in English and Swahili. For the summarization task, we scraped recent news articles in Swahili, extracting a summary from the web page they appeared in. The other two summarization subtasks we planned (cross-lingual summarization and book chapter summarization) did not attract participants, so we do not elaborate on them here. For all tasks,

[1]https://gem-benchmark.com/shared_task

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 17–38
September 23 –27, 2024.

we apply both automatic and human evaluation methods.

In the remainder of this paper, we present the timeline of the task and comment on the incomplete results (Section 2). We then provide an overview of the tasks and datasets involved (Section 3), followed by descriptions of the participating systems (Section 4) and the evaluation methods employed (Section 5). Finally, we present the results available at the time of publication (Section 6).

2 Timeline and status at publication time

The task was advertised in 2023 across different channels, and was officially launched on February 20th 2024, when a pre-registration page was made publicly available. Every team who pre-registered their system was sent the data for the task(s) they selected, with no obligation to submit outputs. All system outputs were collected on April 11th 2024. The following months were dedicated to organising the human evaluation process, and suffered multiple delays, mainly due to the fact that we took a late decision to compile new reference texts for English and Swahili (see Section 5.2).

As a result, at the time of publication of this paper, several evaluations are still ongoing. We only sent the participants the following completed evaluation results: the data-to-text metrics results for English (6 test sets, 7 systems), and the summarization metrics results for Swahili (1 test set, 2 systems). The data-to-text metrics results for Swahili (6 test sets, 3 systems), the human evaluation results for English (6 test sets, 7 systems), Swahili (6 test sets, 3 systems) and Spanish (6 test sets, 3 systems), and the summarization human evaluation results for Swahili (1 test set, 2 systems) are not yet released and are planned to be presented during the INLG conference in September 2024.

3 Overview of tasks

The GEM 2024 shared task consists of two different types of tasks: data-to-text generation and text summarization. Table 1 shows the input/output pairs for each task. Notably, no training or development data was provided to participants for either task. Given the prevalence of large language models, our primary objective was to design test data that was previously unseen by these models. To achieve this, we carefully crafted separate test sets for both the data-to-text and summarization tasks, which are described in detail in this section.

Task	**Input**	**Output**
Data-to-text	Table	Text
Summarization	Full text	Short summary

Table 1: Input/output specifications for the tasks.

3.1 Data-to-text task

The data-to-text (D2T) task consists in generating texts from input triple sets in the WebNLG fashion, where each triple is made of *Subject* | *Property* | *Object*. Figure 1 shows a sample triple set that contains 2 triples (i.e., of size 2). Both triples are about Nie Haisheng (the *Subject*); the first one states his birth date (1964-10-13), while the second one states his occupation (fighter pilot). The expected output in English would be one or two sentences such as "*Nie Haisheng is a fighter pilot born on October 13th 1964*" or "*Nie Haisheng, who was born on October 13th 1964, was a fighter pilot*".

The GEM data-to-text task contains 2 subtasks:

- WebNLG-based (D2T-1): We use the official WebNLG 2020 test set (Castro Ferreira et al., 2020); even though the WebNLG test set contains properties and entities not seen in the training/dev data, we consider the whole WebNLG dataset as in-domain since all splits (training/dev/test) had been available online for more than 3 years before the GEM task was launched. The dataset contains 220 different DBpedia properties and the original dataset specifications can be found on the WebNLG website.[2]

- Wikidata-based (D2T-2): We queried Wikidata to collect 1,800 triples sets containing between 2 and 7 properties for a random set of persons, following the method described in Axelsson and Skantze (2023). The dataset contains 74 different properties, none of which were in WebNLG; furthermore, almost none of the entities are in WebNLG either, so the Wikidata-based tests are considered out-of-domain.[3]

For each subtask, there are 3 parallel test sets, as proposed in Axelsson and Skantze (2023):

[2] https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2020/

[3] Note that the vocabulary of properties of DBpedia and Wikidata are different, but 17 of the 74 Wikidata properties have a direct equivalent with a DBpedia property, e.g., *Occupation/occupation* in Figures 1 and 3.

```
<entry category="Astronaut" eid="Id2" shape="(X (X) (X))" size="2">
  <modifiedtripleset>
    <mtriple>Nie_Haisheng | birthDate | 1964-10-13</mtriple>
    <mtriple>Nie_Haisheng | occupation | Fighter_pilot</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 1: WebNLG Factual input (D2T-1-FA)

```
<entry category="Astronaut" eid="Id2" shape="(X (X) (X))" size="2">
  <modifiedtripleset>
    <mtriple>Martial | birthDate | 1942-01-01</mtriple>
    <mtriple>Martial | occupation | military_engineer</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 2: WebNLG Counterfactual input (D2T-1-CFA)

```
<entry category="Astronaut" eid="Id2" shape="(X (X) (X))" size="2">
  <modifiedtripleset>
    <mtriple>Chryse_Folee | birthDate | May_28_1988</mtriple>
    <mtriple>Chryse_Folee | occupation | Megamace_Trooper</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 3: WebNLG Fictional input (D2T-1-FI)

```
<entry category="WikiData human" eid="Id9" shape="unknown" size="2">
  <modifiedtripleset>
    <mtriple>Bramantino | Occupation | architect</mtriple>
    <mtriple>Bramantino | PlaceOfBirth | Milan</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 4: Wikidata Factual input (D2T-2-FA)

```
<entry category="WikiData human" eid="Id9" shape="unknown" size="2">
  <modifiedtripleset>
    <mtriple>Lambert_of_Ardres | Occupation | politician</mtriple>
    <mtriple>Lambert_of_Ardres | PlaceOfBirth | Umeå</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 5: Wikidata Counterfactual input (D2T-2-CFA)

```
<entry category="WikiData human" eid="Id9" shape="unknown" size="2">
  <modifiedtripleset>
    <mtriple>Chryse_Folee | Occupation | Horizon_Stitcher</mtriple>
    <mtriple>Chryse_Folee | PlaceOfBirth | Oscasala</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 6: Wikidata Fictional input (D2T-2-FI)

- Factual (FA): The information in these inputs is factually correct. For the WebNLG-based task, this test set is the one used for the WebNLG 2020 shared task (Castro Ferreira et al., 2020). Figures 1 and 4 show sample inputs for the D2T-1-FA and D2T-2-FA subtasks respectively.

- Counterfactual (CFA): Entities in the factual dataset are switched based on their Wikidata class (e.g., a person entity is replaced by another person entity, a date by another date, etc.). Figures 2 and 5 show counterfactual inputs derived from Figures 1 and 4, respectively; the properties are the same as in the FA and FI datasets of the subtask (see FI below), but the Subject and Object values are replaced by other existing ones of the same category. In Figure 2, for instance, the information about Marcus Valerius Martialis, known in English as Martial, is factually wrong: Martial was a Roman poet born between 38 and 41 AD. The category feature may not match the new data, but the shape is correct as it is the same as in the original data.

- Fictional (FI): Entities in the factual datasets are replaced by made up entities (obtained via LLM prompting). Figures 3 and 6 show fictional inputs derived from Figures 1 and 4, respectively. In Figure 6 for instance, both the Subject (*Chryse_Folee*) and Object (*Oscasala* and *Horizon_Stitcher*) values are fictional; the properties are the same as in the other 2 subtask datasets (FA and CFA). There is no shape available. The same fictional name appears in the WebNLG example in Figure 3 and the Wikidata example in Figure 6—the same fictional entities may appear several times in different contexts and are not supposed to represent a coherent narrative about anything or anyone.

3.2 Summarization task

Text summarization is the task of producing a concise text sequence that captures the key information from a longer input text. The GEM summarization (Summ) task focuses on news article summarization. We follow the data collection pipeline of XL-Sum (Hasan et al., 2021) to create the task data. The articles are collected from the BBC website.[4] The summaries are extracted from the leading bold paragraph in the web pages containing the news articles, which summarizes the article's information in one or two sentences. To minimize the risk of potential data contamination, we only collect articles published between 2023 and 2024. We collect 2,978 articles in total in English, Spanish, and Swahili. Since all the submissions to the summarization task were in Swahili, we only conducted human evaluation with this subset, where 100 examples were sampled for the evaluation.

3.3 Languages

While the summarization task focused on Swahili, we encouraged submissions in multiple languages for data-to-text, namely Arabic (ar), English (en), Chinese (zh), German (de), Hindi (hi), Korean (ko), Russian (ru), Spanish (es) and Swahili (sw), and told the participants that a subset of these languages

[4] https://www.bbc.com/

Team	D2T-1	D2T-2	Summ	Languages
CUET_SSTM (Rahman et al., 2024)			x	sw
DCU-ADAPT-modPB (Osuji et al., 2024)	x			en, hi, ko, sw
DCU-NLG-PBN (Lorandi and Belz, 2024)	x	x		ar, de, en, es, hi, ko, ru, sw, zh
DCU-NLG-Small (Mille et al., 2024)	x	x		ar, de, en, es, hi, ko, ru, sw, zh
DipInfo-UniTo (Oliverio et al., 2024)	x	x		en
OSU CompLing (Allen et al., 2024)	x	x		en , es
RDFpyrealb (Lapalme, 2024)	x	x		en
SaarLST (Jobanputra and Demberg, 2024)	x	x		en

Table 2: Overview of participating systems

only would be used in the human evaluation, depending on the number of submissions for each (see Section 5.1). The inputs were exactly the same for all the output languages, that is, we did not provide DBpedia triples in Swahili to serve as input the the generation in Swahili; instead, inputs with English labels as in Figures 1 to 6 were used.

4 Participating systems

About 40 teams pre-registered, and 9 submitted outputs; one team eventually withdrew their submission. Table 2 lists the final teams and the subtask(s) and language(s) they addressed. The three DCU teams submitted multiple systems but were asked to choose a primary system for the human evaluation; for the sake of clarity we only report metrics scores for the primary systems, and point the reader to the respective system description papers in this volume for more details about non-primary submissions.

Pre-registration After handing out a preliminary survey to collect interest in the tasks and languages for the shared task, we asked all registered teams to carry out a pre-registration of their planned experiment(s). The objective of the pre-registration is to log in the details of a specific experiment before it is carried out; it is an important initial step to guarantee that the experiment is conducted fairly, and to help avoid potential biases derived from the researchers' interest (van Miltenburg et al., 2021). We asked participants to pre-register selected information (i.e. intended systems, hardware, additional data, automatic metrics, etc.) through a Qualtrics form (see Appendix E for screenshots of the form).

In the following, the summarization baseline and the team submissions are briefly described; an overview of participation to the tasks is provided in Table 2.

The Summarization baseline uses GPT-3.5 following the prompt design from Goyal et al. (2022). The specific prompt is "*Summarize the above article briefly in 1 sentence*" translated into Swahili, "*Fanya muhtasari wa kifungu kilicho hapo juu kwa kifupi katika sentensi 1.*". The system prompt is the default. All output is checked for language id to ensure that the output is in Swahili.

CUET_SSTM (Rahman et al., 2024) uses an integrated extractive-abstractive summarizer. For the extractive summarizer, the authors used the BERT Extractive Summarizer, which shortens long texts of more than 512 tokens. For the abstractive summarizer, they used two pre-trained models (T5-Small, mBART-50) to generate the summaries. The integrated model is trained on the XLSUM Swahili dataset combined with 1,000 manually summarized texts from the given Swahili news classification dataset.

DCU-ADAPT-modPB (Osuji et al., 2024) adopts an NLG+MT approach based on a pipeline neural architecture. It leverages the fine-tuned Flan-T5-large model for the ordering and structuring of input triples. Additionally, a GPT-4 prompt-based model was integrated for surface realisation, generating the final text outputs and employing few-shot prompting with five examples for the final text generation tasks in English. For multilingual text generation in Korean, Arabic, and Swahili, a prompt-based model—the Cohere-command-r-plus neural machine translation model—was incorporated, also using five examples for the translation. For Hindi, the IndicTrans2 model was used.

DCU-NLG-PBN (Lorandi and Belz, 2024) fine-tuned the Mistral 7B Instruct model, using Low-Rank Adaptation (LoRA) to enhance performance while maintaining computational efficiency. The system generates text in English, which is then translated into multiple languages (Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic) using a machine translation system (Google Translate).

DCU-NLG-Small (Mille et al., 2024) combined the FORGe rule-based generator and a post-processing step with a T5-Base model fine-tuned on a parallel dataset of English rule-based-generated texts and human- or LLM-produced texts. For languages other than English, they used the off-the-shelf machine translation system NLLB, which is freely available on HuggingFace.

DipInfo-UniTo (Oliverio et al., 2024) focuses on English and employs a three-step pipeline called the SGA (split-generate-aggregate) pipeline to generate verbalizations. The process begins with a data unit splitting phase, where the initial triples are divided into subsets of three or fewer triples, with an effort to maintain the relationships between them. The next step involves generating verbalizations for each subset of triples using Mistral-7B, which has been fine-tuned on a training and development set from WebNLG 3.0 dataset for English. Finally, in the last step, a pre-trained Mistral-7B model is used for sentence aggregation with a zero-shot prompting technique, merging the generated sentences into a more fluent and coherent text.

OSU CompLing (Allen et al., 2024) experimented with a data filtering and knowledge distillation approach for English, Spanish, Chinese, and English. They leverage the expertise of ChatGPT (GPT 4.0) to generate training data for factual, counterfactual, and fictional triple sets. Data filtering was done with automatic model judgments for error detection. Spanish and English filtered synthetic data was used to fine-tune Llama2.

RDFpyrealb (Lapalme, 2024) employs a symbolic method to address the English D2T challenge. One objective is to contrast the outcomes of computationally demanding techniques that may not always be easy to control with a streamlined, swift, and reliable symbolic method. The design is straightforward: every RDF triple represents a statement, where the subjects and objects of the triple are nearly identical to those of the sentence. The predicate in the triple represents a verb phrase that defines the sentence's syntax. The narrative-building mechanism arranges predicates sequentially, giving rise to a coherent tale. It also combines sentence components when they share the same subject or predicate. The final realization is performed using pyrealb, a French-English realizer which is used in some data to text applications.

SaarLST (Jobanputra and Demberg, 2024) employs a retrieval-augmented generation (RAG) pipeline to generate verbalization. Most RAG pipelines use a dense retriever while this pipeline contains a symbolic retriever – `PropertyRetriever`. The `PropertyRetriever` leverages available WebNLG training and validation sets to retrieve instances with the most similar properties. The retrieved examples and prompting instructions combined form the final few-shot prompt. In the final verbalization step, the pipeline prompts an ensemble of `Mixtral` and `Command-R` models to generate coherent verbalization.

5 Evaluation methods

In Section 3, we detailed the procedure for creating the inputs used in both the D2T and Summ tasks. Initially, these inputs lacked corresponding reference texts. Due to the significant time and resource investments required to create input-output pairs, we strategically delayed collecting human references until we had identified the languages submitted by participants. This section first provides an overview of the language selection and the reference text creation procedures, and then describes the automatic and human evaluations we ran on each submission to the shared task.

5.1 Selection of evaluated languages

As shown in Table 2, for the D2T task, all team submitted English outputs, 3 teams submitted Spanish outputs (DCU-NLG-PBN, DCU-NLG-Small and OSU CompLing), and 3 teams submitted Hindi, Korean and Swahili outputs (DCU-ADAPT-modPB, DCU-NLG-PBN and DCU-NLG-Small); only the two DCU-NLG teams submitted outputs for all other languages. For the Summ task, the only participating team submitted Swahili outputs. The task budget allowed for carrying out human evaluations in 3 languages, and our original plan was to include English and at least one low-resource language. We selected English and Swahili because they had the most submissions, and Spanish to include an additional team in the human evaluation of a language other than English. For English and Swahili, we carry out both automatic and human evaluations, whereas for Spanish, we rely solely on human evaluation.

5.2 Creation of new reference texts in English and Swahili

As mentioned in Section 3, the inputs for both the data-to-text and the summarization tasks have been collected specifically for the present task. Since we recruited bilingual Swahili-English speakers in person for the evaluation of Swahili texts, we also asked them to write reference texts in these two languages for all the D2T test set inputs; there are in total 1,080 input (180 inputs sampled from each of the 6 test sets, see Section 5.4.2), and one text was collected for each input.

The annotators were provided (i) a one-page document with instructions, and (ii) a document with definitions of the 211 different properties found in the sampled test sets, which we drafted ourselves.[5] One meeting with the task organisers and the annotators took place where questions could be asked, and during which the annotators collectively wrote and discussed English and Swahili texts for about 10 input tables. For each English/Swahili text pair created, each annotator received $0.5.

To collect the texts, we used a variation of the evaluation interface (see Section 5.4.4) in which instead of ratings, annotators were shown 2 boxes, one the text in each language. Packages of 12 to 18 input tables were created, and annotators (i) downloaded a package, (ii) submitted the texts for all inputs of the package, and (iii) then had the possibility to download another package not yet used by anyone. For quality control, we collected 2 annotations from different persons for 60 texts.

Due to some delays, we were not able to complete the collection of the above-mentioned texts by the time of publication of this paper. We launched a last-minute set of tasks on Amazon Mechanical Turk (AMT) and Prolific to get the English texts, using some of the English evaluators recruited as described in Section 5.4.1. These are the reference texts we use in the evaluations of the present paper; Appendix C contains a brief assessment of the quality of the collected texts.

5.3 Automatic evaluation

For the **D2T** task, we use a classic set of reference-based metrics for English and Swahili outputs, taking as reference the texts collected as described in Section 5.2; the six D2T test sets contain 180 input/reference pairs each (180 inputs, one reference per input, see Section 5.4.2). The metrics include BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), chrF++ (Popović, 2017) and BERTScore (Zhang et al., 2019). To easily run the evaluation on any pair of predicted and reference files, we released a Notebook[6] largely based on the original WebNLG 2020 code.[7]

For the **Summ** task, we used the BBC automatically generated summaries following the procedure used in the XLSum task. While there were only 200 human evaluations, we use the entire 2,993 test set for the evaluation of the Swahili summarization task. These are several sentences long and provide a baseline summary. Since there were quality issues in the automatically extracted reference summaries, we performed data filtering to resolve these issues, which resulted in 1,367 examples in total. The metrics include ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019) and BARTScore (Yuan et al., 2021).

5.4 Human evaluation

In addition to the automatic evaluations, we also asked human raters to evaluate a subset of the outputs from each submission to the shared task. In this section, we provide details on the evaluator recruitment and training processes, the data sampling, and the evaluation criteria and task design.

5.4.1 Recruitment and training of evaluators

To ensure alignment between the recruited evaluators and the D2T task, we designed a qualification task that consisted of five rating checks and one attention check. For each rating check a handcrafted tabular set of data predicates was presented alongside a text generated from the table. In four of the five rating checks, the text presented contained deliberate errors such as issues with fluency, grammatically, omissions, and additions. Evaluators were asked to assess each text on a 7-point Likert scale on four quality criterion: *fluency*, *grammatically*, *no-additions*, *no-omissions*. In the case of rating checks with deliberate errors for specific quality criterion, evaluators were expected to rate these criteria either neutral (4-rating) and/or lower than neutral. Unaffected aspects were to be rated as higher than neutral. The fifth rating check contained no issues, so evaluators were expected to rate all quality criteria neutral or above.

For the recruitment of English and Spanish evaluators in the D2T task, we used Zhang et al.'s (2023) *qualification task*, where the evaluator is expected to successfully complete a task after receiving a short training. We recruited 23 evaluators in English (15%) pass rate) and 13 in Spanish (22% pass rate) respectively on MTurk and Prolific.

[5] https://github.com/mille-s/GEM24_D2T_StratifiedSampling/tree/main/documents

[6] https://github.com/mille-s/WebNLG-2020_Metrics

[7] https://github.com/WebNLG/GenerationEval.git

Task	Criterion name	Quality type	Frame of reference	Aspect
Table to text	No-Omissions	Correctness	Relative to input	Content
	No-Additions	Correctness	Relative to input	Content
	Grammaticality	Correctness	Output in its own right	Form
	Fluency	Goodness	Output in its own right	Form and Content
Summarization	Understandability	Goodness	Output in its own right	Form and Content
	Faithfulness	Correctness	Relative to input	Content
	Saliency	Goodness	Relative to input	Content
	Grammaticality	Correctness	Output in its own right	Form
	Coherence	Goodness	Output in its own right	Content
	Compactness	Goodness	Output in its own right	Content

Table 3: Properties of our criteria according to the taxonomy by Belz et al. (2020).

On the other hand, recruiting evaluators from low-resource languages (Swahili in our case) on crowdsourced platforms is more challenging. Thus, for both tasks, we recruited 14 students who are Swahili native speakers from the Technical University of Kenya and Moi University. To help these students understand the task, we (i) set up meetings to explain the task in detail, (ii) carried out a few tasks together, and (iii) formed a Google group for questions and discussion.

5.4.2 Data sampling and packaging

For the **D2T** task, we selected 180 data points (∼10%) from each of the six test sets (D2T-1-FA, D2T-1-CFA, D2T-1-FI, D2T-2-FA, D2T-2-CFA, D2T-2-FI, see Section 3), stratifying only by input size and excluding inputs of size 1, which are usually trivial to generate from. Thus, each of the six test sets contains 30 inputs for each input size, ranging from 2 to 7. This allows us to analyze the metrics results broken down by input size. The code for sampling and creating the corresponding pairs of HTML tables and system outputs as used in the human evaluation is available on GitHub.[8]

Once sampled, the input/output pairs were packaged to be sent to the evaluators. For Swahili, we created 75 packages of 36 input/output pairs. For Spanish, we created 270 packages of 12 input/output pairs. For English, we created 1,080 packages of 7 - 8 input/output pairs. The packages for English and Spanish are substantially smaller that those for Swahili because the evaluators for these two languages were recruited on Amazon Mechanical Turk, where proposed tasks are usually short. The Swahili evaluators were recruited in person and could be trusted to complete larger packages.[9]

The **Summ** outputs were not sampled nor packaged at the time this paper was written.

5.4.3 Quality criteria

The criteria used for the evaluation should capture aspects of the quality of the meaning and form. Table 3 lists the criteria used in both tasks and lists their properties according to Belz et al.'s (2020) taxonomy.

D2T task Our selection of criteria (see Table 4) reflects closely the evaluations carried out in the context of some recent data-to-text shared tasks such as WebNLG (Cripwell et al., 2023) or E2E (Dušek et al., 2020). We evaluated four dimensions, namely whether or not the text represents faithfully the input table (*No-Omissions*, *No-Additions*), whether or not the text contains grammatical errors (*Grammaticality*), and whether or not the output text flows well on its own (*Fluency*).

Criterion name	Definition
No-Omissions	ALL the information in the table is present in the text.
No-Additions	ONLY information from the table is present in the text.
Grammaticality	The text is free of grammatical and spelling errors.
Fluency	The text flows well and is easy to read; its parts are connected in a natural way.

Table 4: Criteria used for data-to-text generation

Summ task The objective of the evaluation is to assess the quality of a summary given an input text. The summaries are evaluated along the dimensions defined in Zhang et al. (2023), shown in Table 5 with their respective definitions. The objective of

[8] https://github.com/mille-s/GEM24_D2T_StratifiedSampling. Thanks to Liam Cripwell and Michel Lorandi for making the WebNLG 2023 sampling code available, which we used as a starting point.

[9] The Swahili packages represent about one hour of work; we tried packages of the same size on Mechanical Turk and received complaints from Turkers that the tasks were too long.

Criterion name	Definition
Understandability	Can the worker understand the summary and is the summary worth being annotated.
Faithfulness	All of the information in the summary can be found in the article; the summary accurately reflects the contents of the article.
Saliency	The summary captures the most important information of the article and does not include parts of the article that are less important.
Grammaticality	The summary is free of grammatical and spelling errors.
Coherence	The summary is presented in a clear, well-structured, logical, and meaningful way.
Compactness	The summary does not contain duplicated information.

Table 5: Criteria used for summarization

the first criterion, *Understandability*, is to give the annotator a chance to not provide the ratings for the rest of the criteria in case the quality of the text does not allow for it. Two criteria (*Faithfulness* and *Saliency*) require the evaluators to compare the summary with the input, while the remaining three (*Grammaticality*, *Coherence*, *Compactness*) capture intrinsic qualities of the summary. Two criteria are highly specific to the summarization task, namely *Saliency* and *Compactness*, which aim at capturing respectively whether the main points of the original text were captured, and whether the resulting summary is indeed compact and does not contain unnecessary repetitions.

5.4.4 Survey Design

We designed evaluation surveys for data-to-text and summarization using HTML, CSS, and Jinja. We launched our survey on Amazon Mechanical Turk and Prolific. For all tasks, evaluators were shown the input and one output (see Table 1). For all criteria, direct assessment was used, and the answers were collected using a labeled 7-point scale (see Figure 7). The evaluation interfaces are shown in Figures 8 and 9 in Appendix A.

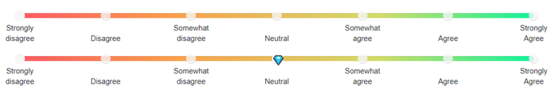


Figure 7: Rating Scale (7-point)

Designing an effective survey requires an understanding of the subject matter and awareness of potential biases that could compromise validity, and we drew our inspiration from HCI research practices (Müller and Sedley, 2015). We aimed to create a reliable and impactful survey by minimising biases and tailoring each aspect to elicit meaningful, accurate responses. See Appendix B for more discussion on the choices behind the survey design.

6 Results

In this section, we present the results of the metrics evaluation for the English data-to-text task and the Swahili summarization task.

6.1 Metrics results for the D2T task

Table 8 shows the **BLEU**, **METEOR**, **chrF++** and **BERT's F1** scores of all primary systems on the three D2T-1 and the three D2T-2 test sets respectively (FA = Factual, CFA = Counterfactual, FI = Fictional, see Section 3) for the English language. For all the results broken down by input size, see the plots in Appendix D. As mentioned above, for calculating the scores in Table 8, we use the references created by our AMT-recruited annotators (see Section 5.2). For comparison, we also report here the scores obtained with the entire WebNLG test set (1,779 texts) and all the WebNLG references (Table 6), and the scores obtained with the same set of 180 data points as in Table 8, but selecting only one random WebNLG reference when more than one is available.[10]

Comparison between the D2T-1 and the D2T-2 subtasks. For all six systems that participated in both subtasks, the scores are substantially higher for the D2T-1 task than for the D2T-2 for the factual (FA) and fictional (FI) datasets, but, surprisingly, not for the counterfactual (CFA) dataset, where scores are always higher in the D2T-2 subtask. For DCU-NLG-PBN, DipInfo-UniTo, OSU-CompLing and SaarLST (i.e. all submissions that are not primarily based on a rule-based system), BERTScore is even equal or higher for all 3 datasets of the D2T-2 task.

D2T-1 scores. All seven submissions obtained a (generally substantially) higher score for all metrics on the factual (FA) dataset, which was expected since this is the only dataset for which reference texts were available when the task was running. For all seven submissions, BERT systematically

[10]The number of references used can affect the scores of some metrics, for example, BLEU.

System ID	BLEU ↑	METEOR ↑	chrF++ ↑	BERT F1 ↑
DCU-ADAPT-modPB	49.8	0.400	0.655	0.955
DCU-NLG-PBN	**52.26**	**0.410**	0.679	**0.956**
DCU-NLG-Small	51.43	0.395	0.662	0.954
DCU-NLG-Small-noT5	40.55	0.372	0.620	0.943
DipInfo-UniTo	51.36	**0.410**	**0.681**	0.955
OSU CompLing	43.09	0.389	0.65	0.950
RDFpyrealb	42.38	0.390	0.642	0.946
SaarLST	39.86	0.400	0.655	0.947

Table 6: Metrics scores on the D2T-1-FA English test set using all WebNLG data points (1,779) and all reference texts (2.5 texts per data point on average).

System ID	BLEU ↑	METEOR ↑	chrF++ ↑	BERT F1 ↑
DCU-ADAPT-modPB	28.27	0.338	0.561	0.936
DCU-NLG-PBN	**32.5**	**0.356**	**0.6**	**0.937**
DCU-NLG-Small	29.17	0.337	0.571	0.933
DipInfo-UniTo	30.47	0.348	0.585	0.93
OSU CompLing	27.01	0.339	0.575	0.931
RDFpyrealb	26.26	0.339	0.567	0.927
SaarLST	25.61	0.354	0.59	0.931

Table 7: Metrics scores on the D2T-1-FA English test set using the 180 data points of the human evaluation and 1 randomly selected WebNLG reference text per data point.

		D2T-1			**D2T-2**		
	System	**FA**	**CFA**	**FI**	**FA**	**CFA**	**FI**
	DCU-ADAPT-modPB	30.78	26.98	26.54	n/a	n/a	n/a
	DCU-NLG-PBN	29.08	25.2	26.02	23.96	30.34	20.46
	DCU-NLG-Small	27.0	22.98	20.85	19.48	24.9	16.88
BLEU ↑	DipInfo-UniTo	**32.31**	**29.01**	**28.24**	27.22	**32.01**	**21.26**
	OSU CompLing	30.03	24.45	21.44	24.97	27.06	16.9
	RDFpyrealb	26.37	21.67	21.97	19.97	25.05	16.28
	SaarLST	29.7	23.48	20.76	**28.25**	26.47	20.16
	DCU-ADAPT-modPB	0.332	0.299	0.318	n/a	n/a	n/a
	DCU-NLG-PBN	0.33	0.297	0.322	0.295	0.348	0.3
	DCU-NLG-Small	0.314	0.279	0.292	0.26	0.3	0.267
METEOR ↑	DipInfo-UniTo	0.346	**0.315**	**0.342**	0.304	0.354	0.307
	OSU CompLing	0.335	0.293	0.306	0.295	0.334	0.282
	RDFpyrealb	0.331	0.291	0.31	0.287	0.335	0.286
	SaarLST	**0.347**	0.307	0.331	**0.32**	**0.359**	**0.315**
	DCU-ADAPT-modPB	0.555	0.515	0.539	n/a	n/a	n/a
	DCU-NLG-PBN	0.555	0.513	0.549	0.49	0.581	0.49
	DCU-NLG-Small	0.537	0.488	0.507	0.438	0.51	0.442
chrF++ ↑	DipInfo-UniTo	0.58	**0.543**	**0.587**	0.512	0.592	0.502
	OSU CompLing	0.566	0.514	0.537	0.496	0.567	0.475
	RDFpyrealb	0.551	0.495	0.527	0.479	0.561	0.472
	SaarLST	**0.581**	0.524	0.557	**0.538**	**0.597**	**0.518**
	DCU-ADAPT-modPB	**0.935**	0.924	0.921	n/a	n/a	n/a
	DCU-NLG-PBN	0.933	0.923	0.92	0.936	**0.937**	**0.924**
	DCU-NLG-Small	0.93	0.918	0.914	0.925	0.923	0.914
BERT F1 ↑	DipInfo-UniTo	0.933	**0.926**	**0.924**	**0.937**	0.936	0.923
	OSU CompLing	0.932	0.92	0.915	0.934	0.93	0.917
	RDFpyrealb	0.928	0.918	0.917	0.921	0.923	0.916
	SaarLST	0.931	0.921	0.917	0.934	0.929	0.919

Table 8: Metrics scores for the English D2T task (180 data points, 1 AMT reference text per data point).

sores the counterfactual (CFA) texts higher than the fictional (FI) texts, while METEOR and chrF++ exhibit the opposite behaviour. BLEU behaves very similarly to BERT.

D2T-2 scores. For all systems except SaarLST, the scores for all metrics on the counterfactual dataset (CFA) are higher than for the other two datasets (FA, FI); for these systems, only BERTScore sometimes gets slightly higher scores for Factual (FA) datasets. BLEU and BERT usually score FA texts clearly higher than fictional (FI) ones, while for METEOR and chrF++, FA and FI texts receive very similar scores.

From the perspective of system submissions, DipInfo-UniTo scores comparatively high for all metrics on all datasets. DCU-NLG-PBN and DipInfo-UniTo seem to degrade less than other systems when comparing the FA scores to the CFA and FI scores for D2T-1; for D2T-2, the submissions using rule-based components (DCU-NLG-Small and RDFpyrealb) have less drop than others from FA to FI (these two also have comparable scores overall). SaarLST seems to be the system that suffers the least when exposed to the out-of-domain data (D2T-2). When comparing the results on the AMT references (Table 8) and the ones with the WebNLG references (Tables 6 and 7), one can note that SaarLST for instance obtains higher scores on the D2T-1-FA dataset with AMT references than on the dataset with WebNLG references, while DCU-NLG-Small, which used a component fine-tuned using BLEU on the WebNLG dataset, obtains higher scores with WebNLG references than with AMT references.

At this point, and without the results of the human evaluation, it is unclear to what extent all the score differences mentioned above are due to the properties of the inputs and outputs, or to some features of the reference texts. A more in-depth analysis of the results will be provided at a later stage along with the human evaluation results.

6.2 Metrics results for the summarization task

We use ROUGE, BARTScore, and BERTScore for the automatic evaluation of the summarization system. For BARTScore, we use a multilingual BART checkpoint introduced in Tang et al. (2020).[11] Similarly, we use a multilingual BERT checkpoint[12] for BERTScore. Apart from the submitted system, we also evaluate a strong baseline that prompts GPT-3.5[13] to generate summaries with one sentence (see Section 4).

The evaluation results are reported in Table 9. CUET_SSTM achieves better performance in ROUGE scores, while GPT-3.5 achieves a higher BARTScore. Regarding BERTScore, CUET_SSTM achieved a higher recall score but a lower precision score, which is correlated with the fact that the average summary length of CUET_SSTM is much smaller. We note that since GPT-3.5's summaries are generated in a zero-shot manner, comparing its summaries using reference-based evaluation metrics may not always be accurate (Goyal et al., 2022; Liu et al., 2023). However, these results indicate that CUET_SSTM is able to achieve a relatively strong performance under the reference-based evaluation.

System	R1	R2	BARTS.	BERTS.	Len.
GPT-3.5	27.12	10.42	-6.305	69.33/73.18	31.10
CUET_SSTM	29.33	15.87	-6.791	71.05/71.37	19.59

Table 9: Automatic evaluation results of the submitted summarization system and the basline. R1 and R2 are ROUGE-1 and ROUGE-2 respectively. BARTS. and BERTS. are BARTScore and BERTScore. Len. is the average number of words in summaries. For BERTScore, we report both the precision/recall scores.

7 Conclusions

We presented an overview of the two tasks of the 2024 GEM shared task, multilingual data-to-text generation and news article summarization in Swahili. For both tasks, we collected new data with the objective provide challenging inputs to the large language models that we supposed most teams were going to use. For the data-to-text task, 7 teams submitted outputs in one or more languages, and we report on the metrics evaluation for English outputs only. The results of the evaluation show that despite the variety of system types (LLMs, rule-based, combination of the two), all systems seem to suffer when exposed to (i) out-of-domain data, and (ii) counterfactual or fictional data. The unexpectedly high scores obtained by all systems on the counterfactual out-of-domain dataset remain to be explained, possibly in the light of the human evaluation results. For the summarization task in Swahili, we received only one submission, which is competitive with a zero-shot GPT-3.5 baseline according to the metrics evaluation.

We were not able to complete all evaluations at the time the paper is published, and the data-to-text metrics results for Swahili, the human evaluation results for English, Swahili and Spanish, and the summarization human evaluation results for Swahili will be reported in a separate publication.

Acknowledgements

We thank Google for funding our crowdsourcing annotations. Mille's contribution was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS).

[11] https://huggingface.co/facebook/mbart-large-50

[12] https://huggingface.co/google-bert/bert-base-multilingual-cased

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

References

Mohd Azry Abdul Malik, Muhammad Firdaus Mustapha, Norafefah Mohamad Sobri, Nor Fatihah Abd Razak, Mohamad Nurifaizal Mohd Zaidi, Ahmad Aizat Shukri, and Muhammad Amir Luqman Zalimie Sham. 2021. Optimal reliability and validity of measurement model in confirmatory factor analysis: Different likert point scale experiment. *Journal of Contemporary Issues and Thought*, 11(1):105–112.

Alyssa Allen, Ash Lewis, Yi-Chien Lin, Tomiris Kaumenova, and Mike White. 2024. OSU Compling at the GEM'24 data-to-text task. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. *arXiv preprint arXiv:2307.07312*.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

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.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, William Soto-Martinez, et al. 2023. The 2023 webnlg shared task on low resource languages overview and evaluation results (webnlg 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2023. Nl-augmenter: A framework for task-sensitive natural language augmentation.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Ron Garland. 1991. The mid-point on a rating scale: Is it desirable? *Marketing Bulletin*, 2.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv

Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Papangelis, Aman Madaan, Angelina Mcmillan-major, Anna Shvets, Ashish Upadhyay, Bernd Bohnet, Bingsheng Yao, Bryan Wilie, Chandra Bhagavatula, Chaobin You, Craig Thomson, Cristina Garbacea, Dakuo Wang, Daniel Deutsch, Deyi Xiong, Di Jin, Dimitra Gkatzia, Dragomir Radev, Elizabeth Clark, Esin Durmus, Faisal Ladhak, Filip Ginter, Genta Indra Winata, Hendrik Strobelt, Hiroaki Hayashi, Jekaterina Novikova, Jenna Kanerva, Jenny Chim, Jiawei Zhou, Jordan Clive, Joshua Maynez, João Sedoc, Juraj Juraska, Kaustubh Dhole, Khyathi Raghavi Chandu, Laura Perez Beltrachini, Leonardo F . R. Ribeiro, Lewis Tunstall, Li Zhang, Mahim Pushkarna, Mathias Creutz, Michael White, Mihir Sanjay Kale, Moussa Kamal Eddine, Nico Daheim, Nishant Subramani, Ondrej Dusek, Paul Pu Liang, Pawan Sasanka Ammanamanchi, Qi Zhu, Ratish Puduppully, Reno Kriz, Rifat Shahriyar, Ronald Cardenas, Saad Mahamood, Salomey Osei, Samuel Cahyawijaya, Sanja Štajner, Sebastien Montella, Shailza Jolly, Simon Mille, Tahmid Hasan, Tianhao Shen, Tosin Adewumi, Vikas Raunak, Vipul Raheja, Vitaly Nikolaev, Vivian Tsai, Yacine Jernite, Ying Xu, Yisi Sang, Yixin Liu, and Yufang Hou. 2022. GEMv2: Multilingual NLG benchmarking in a single line of code. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 266–281, Abu Dhabi, UAE. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Mayank Jobanputra and Vera Demberg. 2024. TeamsaarLST at the GEM'24 data-to-text task: Revisiting symbolic retrieval in the LLM-age. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Zdeněk Kasner and Ondřej Dušek. 2024. Beyond reference-based metrics: Analyzing behaviors of open llms on data-to-text generation. *arXiv preprint arXiv:2401.10186*.

Elena Klyachko, Alexey Sorokin, Natalia Krizhanovskaya, Andrew Krizhanovsky, and Galina Ryazanskaya. 2020. Lowresourceeval-2019: a shared task on morphological analysis for low-resource languages. *Preprint*, arXiv:2001.11285.

Guy Lapalme. 2024. RDFPYREALB at the GEM'24 data-to-text task: Symbolic english text generation from RDF triples. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

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.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Michela Lorandi and Anya Belz. 2024. DCU-NLG-PBN at the GEM'24 data-to-text task: Open-source LLM PEFT-Tuning for effective data-to-text generation. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. 2021. Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the HuggingFace and GEM data and model cards. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 121–135, Online. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia. Association for Computational Linguistics.

Simon Mille, Kaustubh D. Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic construction of evaluation suites for natural language generation datasets. *ArXiv*, abs/2106.09069.

Simon Mille, Mohammed Sabry, and Anya Belz. 2024. DCU-NLG-Small at the GEM'24 data-to-text task: Rule-based generation and post-processing with T5-base. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Hendrik Müller and Aaron Sedley. 2015. Designing surveys for hci research. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 18.

Marcel Nawrath, Agnieszka Nowak, Tristan Ratz, Danilo Walenta, Juri Opitz, Leonardo Ribeiro, João Sedoc, Daniel Deutsch, Simon Mille, Yixin Liu, Sebastian Gehrmann, Lining Zhang, Saad Mahamood, Miruna Clinciu, Khyathi Chandu, and Yufang Hou. 2024. On the role of summary content units in text summarization evaluation. 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 272–281, Mexico City, Mexico. Association for Computational Linguistics.

Atul Kr. Ojha, Chao-Hong Liu, Katharina Kann, John Ortega, Sheetal Shatam, and Theodorus Fransen. 2021. Findings of the loresmt 2021 shared task on covid and sign language for low-resource languages. *Preprint*, arXiv:2108.06598.

Atul Kr. Ojha, Valentin Malykh, Alina Karakanta, and Chao-Hong Liu. 2020. Findings of the LoResMT 2020 shared task on zero-shot for low-resource languages. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 33–37, Suzhou, China. Association for Computational Linguistics.

Michael Oliverio, Pier Felice Balestrucci, Alessandro Mazzei, and Valerio Basile. 2024. DipInfo-UniTo at the GEM'24 data-to-text task: Augmenting LLMs with the split-generate-aggregate pipeline. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Colm O'Muircheartaigh, Jon A. Krosnick, and Armen Helic. 2000. Middle Alternatives, Acquiescence, and the Quality of Questionnaire Data. Working Papers 0103, Harris School of Public Policy Studies, University of Chicago.

Chinonso Cynthia Osuji, Rudali Huidrom, Kolawole John Adebayo, Thiago Castro Ferreira, and Brian Davis. 2024. DCU-ADAPT-modPB at the GEM'24 data-to-text generation task: Model hybridisation for pipeline data-to-text natural language generation. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. 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.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Samia Rahman, Momtazul Arefin Labib, Hasan Murad, and Udoy Das. 2024. CUET_SSTM at the GEM'24 summarization task: Integration of extractive and abstractive method for long text summarization in Swahili language. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Emiel van Miltenburg, Chris van der Lee, and Emiel Krahmer. 2021. Preregistering NLP research. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 613–623, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Lining Zhang, Simon Mille, Yufang Hou, Daniel Deutsch, Elizabeth Clark, Yixin Liu, Saad Mahamood, Sebastian Gehrmann, Miruna Clinciu, Khyathi Raghavi Chandu, and João Sedoc. 2023. A needle in a haystack: An analysis of high-agreement workers on MTurk for summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14944–14982, Toronto, Canada. Association for Computational Linguistics.

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

A Screenshots of evaluator interface

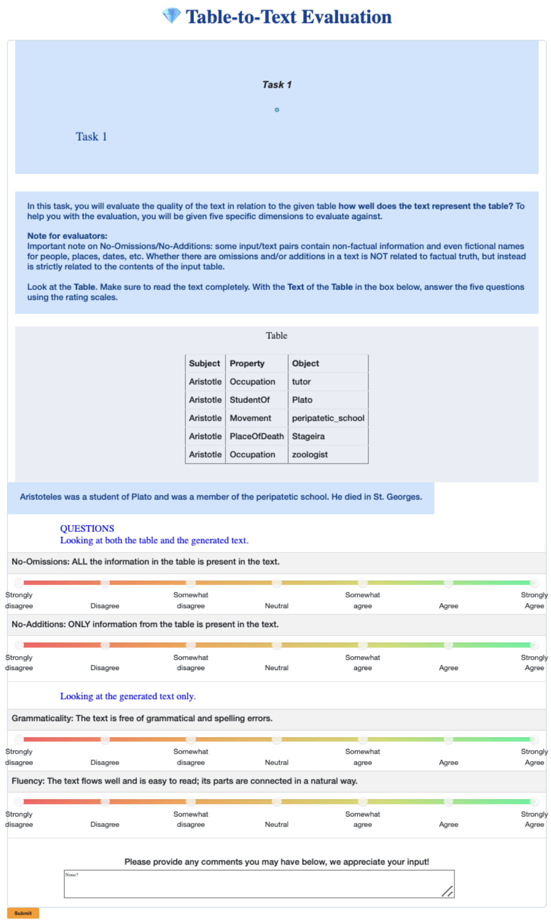


Figure 8: Data-to-text UI

B Justification of the survey design

A 7-point Likert scale offers respondents a broader range of options, enabling evaluators to express their opinions with greater nuance and precision. This expanded scale reduces the likelihood that respondents will default to a middle option out of uncertainty, thereby enhancing the accuracy of the data collected. By providing more choices, a 7-point scale allows for a more accurate reflection of respondents true feelings. Research has demonstrated that increasing the number of points on a Likert scale not only improves the reliability of the data but also reduces the potential for random error (Abdul Malik et al., 2021). On the other hand, closed-ended questions can introduce biases that may affect the data. For instance, the phrasing of questions, the order of response options, and the inclusion of a neutral midpoint can all influence how respondents' interpret and answer questions.

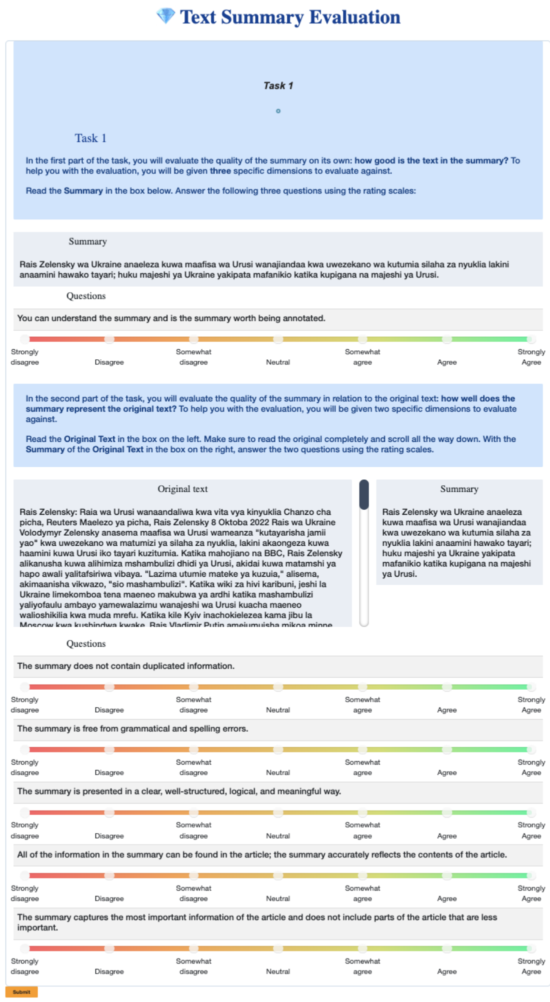


Figure 9: Text summarization UI

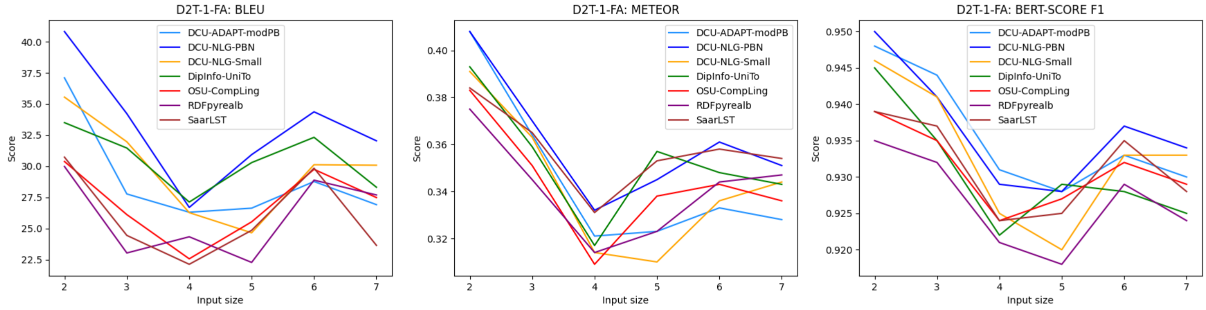
Garland (1991) examined the impact of including or excluding a neutral midpoint on a Likert scale in surveys and found that removing the midpoint can reduce social desirability bias but may push respondents toward more extreme ratings, potentially distorting results. This highlights the need to carefully consider the inclusion of a midpoint, as it can significantly influence survey outcomes. Later, O'Muircheartaigh et al. (2000) found that offering a middle alternative reduces random measurement error, increasing the reliability of responses without affecting validity. Contrary to concerns, their study suggests that including a midpoint improves data quality and does not increase acquiescence bias. Therefore, we decided to include the midpoint as "Neutral" 💎, as presented in Figure 7.

C Informal assessment of the quality of English texts collected on AMT

While collecting texts on AMT, the authors applied manual and automatic filters. When the 1,080 (180*6) final texts were collected, one of the authors of the present paper selected randomly about 10 texts for each of the 6 datasets (60 texts in total), and checked whether or not the texts were adequately verbalising their respective input table. For 20 of these texts (1/3), some problems were detected, such as omissions, nonsensical contents, pasting irrelevant text, additions, or inaccurate verbalisation of some triples (e.g. inversions of Subject and Object or wrong semantics of the property). Additions are being noticed in particular (but not only) on the factual data, suggesting that some workers used language models to create the texts despite clear instructions not to do so. The rest of these texts (2/3) were judged of excellent quality.

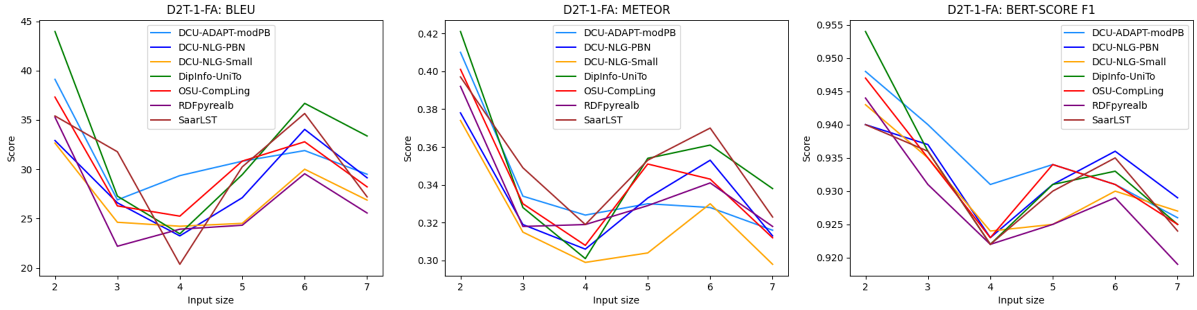
D English D2T metrics evaluation broken down by input size

Figures 11 to 16 show the plots of the results in Table 8, broken down by input size (from size 2 to size 7). Figure 10 shows the same using the WebNLG references (i.e. for the D2T-1-FA dataset), for comparison.



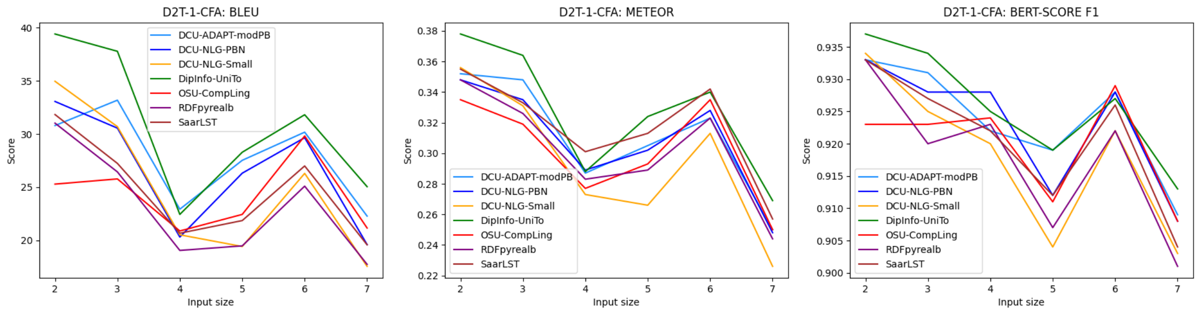
(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 10: Metrics scores per input size (D2T-1-FA) using one randomly selected original WebNLG reference for the 180 sampled data points used in the human evaluation.



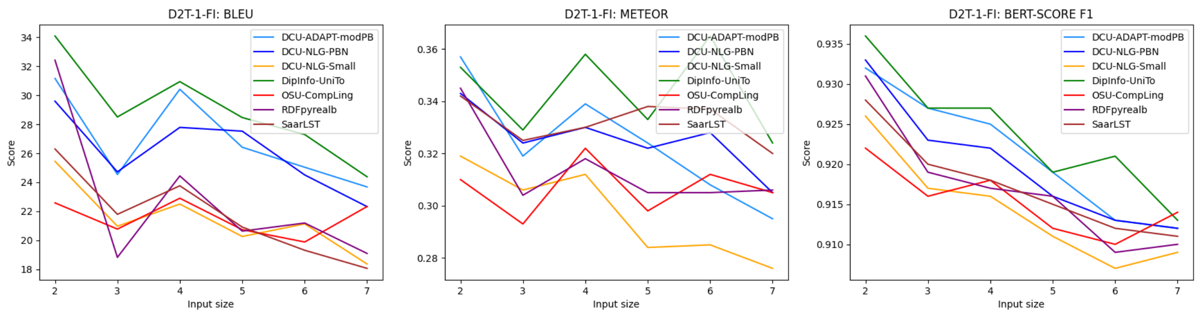
(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 11: Metrics scores by input size on the D2T-1-FA English task (1 AMT reference text per data point)



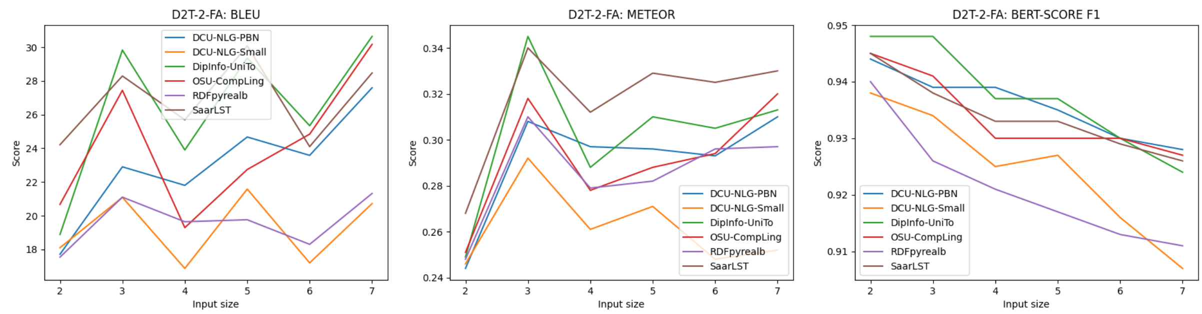
(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 12: Metrics scores by input size on the D2T-1-CFA English task (1 AMT reference text per data point)



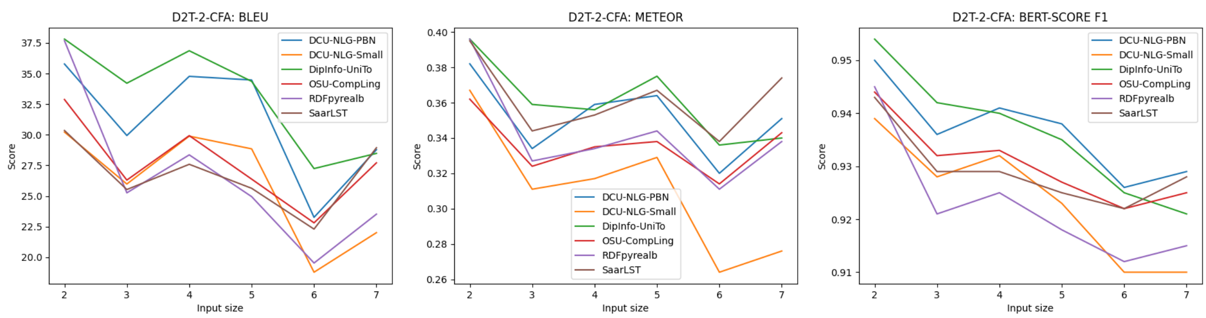
(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 13: Metrics scores by input size on the D2T-1-FI English task (1 AMT reference text per data point)



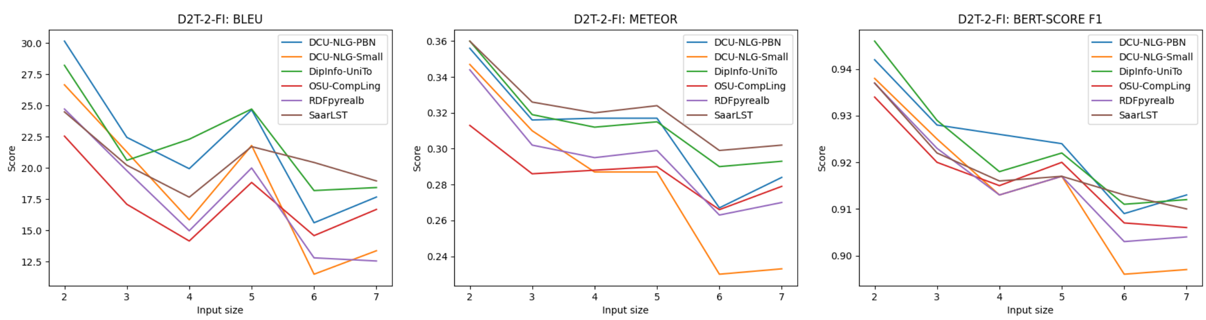
(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 14: Metrics scores by input size on the D2T-2-FA English task (1 AMT reference text per data point)



(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 15: Metrics scores by input size on the D2T-2-CFA English task (1 AMT reference text per data point)



(a) BLEU (b) METEOR (c) BERTScore-F1

Figure 16: Metrics scores by input size on the D2T-2-FI English task (1 AMT reference text per data point)

E Pre-registration Form

For details about the pre-registration form, please see the file below.

Default Question Block

Team Name

Team leader's name

Team leader's email (preferably an institutional email)

Team leader's research group / organization

Team leader's affiliation

Team members (separate each member with semicolons: name1, email1; name2, email2; ...)

Please specify your system name (system name in case of multiple systems for one team)

Block 1

Pre-registration questions

[Read the documentation about the shared task here.]

What is your intended system(s) that you plan to use for the task(s)?
(e.g., Fine-tuned with parameter efficient fine-tuning using LLAMA-2 7B with a multi-step inference.)

Do you have any specific details that you would like to pre-register?
(e.g., We will pre-train using the XLSum dataset and possibly an internal dataset of 100 tailored examples. We may also use in context learning to prompt engineer solutions. Finally, we may also use GPT-4 to create fine-tuning examples for our model.)

What software libraries will you use?

(e.g., Pytorch Huggingface library)

What hardware will you use?
(e.g., Azure server with 8 X A100 80 GB)

What parameter settings will you use?
(e.g., LLAMA-7B 8-bit fine-tuning)

Do you plan to use additional data? What are its key properties?
(e.g., We will use ShareGPT data)

Will you use automatic metric(s)? If yes, which metric(s) (including implementation) will you use, and how will they be configured?
(e.g., We will use G-Eval for automatic analysis.)

Will you carry out an error analysis?
(e.g., We will manually examine the output in order to verify the model and the prompt engineering.)

Anything else you'd like to preregister?

Which Data-to-Text and Summarization subtasks are you planning to submit to

- [ ] Data-to-Text Subtask 1: WebNLG-based (D2T-1)
- [ ] Data-to-Text Subtask 2: Wikidata-based (D2T-2)
- [ ] Summarization Subtask 1: Underrepresented Language Summarization (Swahili)
- [ ] Summarization Subtask 2: Cross-lingual Summarization
- [ ] Summarization Subtask 3: English Book Chapter Summarization
- [ ] I don't know yet

Summary of the Visually Grounded Story Generation Challenge

Xudong Hong[13], **Asad Sayeed**[2], **Vera Demberg**[13]

[1]Dept. of Language Science and Technology and Dept. of Computer Science, Saarland University
[2]Dept. of Philosophy, Linguistics, and Theory of Science, University of Gothenburg
[3]Saarland Informatics Campus, Saarbrücken
{xhong,vera}@lst.uni-saarland.de
asad.sayeed@gu.se

Abstract

Recent advancements in vision-and-language models have opened new possibilities for natural language generation, particularly in generating creative stories from visual input. We thus host an open-sourced shared task, Visually Grounded Story Generation (VGSG), to explore whether these models can create coherent, diverse, and visually grounded narratives. This task challenges participants to generate coherent stories based on sequences of images, where characters and events must be grounded in the images provided. The task is structured into two tracks: the Closed track with constraints on fixed visual features and the Open track which allows all kinds of models. We propose the first two-stage model using GPT-4o as the baseline for the Open track that first generates descriptions for the images and then creates a story based on those descriptions. Human and automatic evaluations indicate that: 1) Retrieval augmentation helps generate more human-like stories, and 2) Large-scale pre-trained LLM improves story quality by a large margin; 3) Traditional automatic metrics can not capture the overall quality.[1]

1 Introduction

Vision-based language generation (VLG) is the generation of text from visual input and is an important task in natural language generation and artificial intelligence. Recently, large pre-trained vision-and-language models (VLMs), such as GPT-4 (OpenAI, 2023) and Gemini (Reid et al., 2024), have achieved remarkable performance across several multimodal tasks, including image captioning (Vinyals et al., 2016), visual question answering (Goyal et al., 2017), and visual dialogue generation (Das et al., 2017).

Although these advancements are notable, most of the current tasks involve predicting labels or generating short pieces of text (typically under 30 words). It remains uncertain whether the latest VLMs can create longer, coherent texts consisting of multiple sentences based on visual input. The evaluation of long stories is still challenging (Min et al., 2023). On the other hand, humans can easily generate extended and logically connected text from visual stimuli. To further assess VLMs, a task more aligned with human capabilities is necessary (Bubeck et al., 2023).

Previous tasks have been designed to evaluate the ability of VLMs to produce more extended outputs, such as visual paragraphs (Krause et al., 2017), localized narratives (Pont-Tuset et al., 2020), and video captioning (Voigtlaender et al., 2023). However, these tasks primarily focus on literal descriptions, where sentences remain independent rather than forming a coherent whole. Coherence, especially local coherence—defined as the relationships between entities in a given context—is fundamental to human language comprehension and production. In vision and language research, local coherence is crucial for several reasons: **1.** Improved models of local coherence can enhance the performance of vision-language tasks, such as text-to-image retrieval (Park and Kim, 2015). **2.** Accurately modeling coherence is essential for developing event knowledge, as events revolve around entities. Stronger event modeling enhances vision-language pre-training (Zellers et al., 2021, 2022).

Story generation is a widely researched task in natural language generation and is frequently used to assess whether large pretrained models can track entities (Paperno et al., 2016) and produce locally coherent texts. Unlike image captions, stories involve multiple characters and events, with recurring entities interacting with one another and their surroundings. Moreover, the importance of characters and relevant content is central to successful story creation (Goldfarb-Tarrant et al., 2020). We contend that story generation is an appropriate bench-

[1]Source code and pre-trained models are available at https://vgsg2024.github.io/

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 39–46
September 23 –27, 2024. ©2024 Association for Computational Linguistics

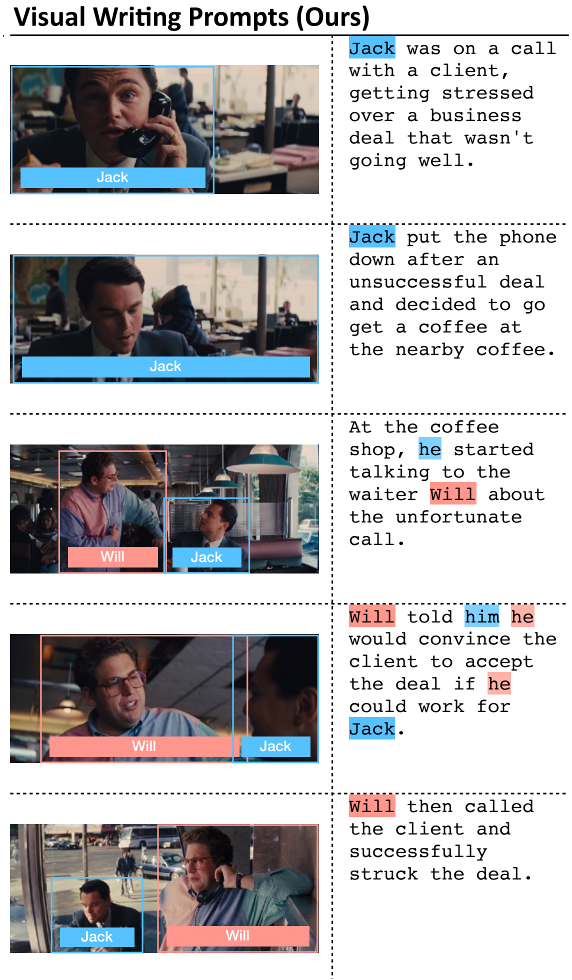


Figure 1: Example of Visual Grounded Story Generation on Visual Writing Prompts dataset. The dataset has recurring characters across all five images and sub-stories. Each occurrence of a character in a sub-story has a bounding box in the corresponding image, which grounds the textual appearance to visual input.

mark for testing the ability of VLMs to generate coherent text.

In response, we introduce a new shared task called Visually Grounded Story Generation (**VGSG**), which challenges VLMs to generate coherent, diverse, and visually grounded stories. This task presents two primary challenges: **1.** The characters in the stories must be grounded in the images, meaning their actions and descriptions should align with the visual information provided. **2.** The generated stories must be coherent, with a clear beginning, middle, and end, and maintain a logical progression from one sentence to the next. Our goal is to Identify the pros and cons of the current VLMs and automatic metrics on this task.

We conduct both automatic and human evaluations. For automatic evaluations, we mainly employ traditional metrics, including BLEU scores (B; Papineni et al., 2002), METEOR (M; Banerjee and Lavie, 2005), ROUGE-L (R; Lin, 2004), and CIDEr (C; Vedantam et al., 2015), to set up an efficient standard evaluation pipeline for this task. We also follow Hong et al. (2023b) to create a solid human evaluation across properties for good stories including Coherence, Diversity, Grammaticality, Visual Grounding, and Overall quality.

Our major findings are 1) Retrieval augmentation based on visual input similarities aids in generating more human-like stories; 2) Large-scale pre-trained language models significantly enhance story quality in that proprietary models with large-scale pre-training are still difficult to outperform using smaller models; and 3) Traditional automated metrics are inadequate in assessing overall quality because they do not correlate with human judgments.

Through this shared task, we hereby call for further research on visually grounded story generation, especially on the evaluation of the excessively long output from the models with large-scale pre-training.

2 Task Description

We define the VGSG task as follows: given a sequence of images (like the first column of Figure 1) the system needs to generate a coherent short story conditioned on the image sequence (like the second column of Figure 1). In addition, the generated story should contain the characters seen in the image sequence.

The VGSG shared task focuses on coherent and visually grounded stories with high diversity.

2.1 Datasets

We use four datasets for evaluation, two of which provide grounding annotations for characters. One of these is our own Visual Writing Prompts dataset:

Visual Writing Prompts (VWP; Hong et al., 2023b), a vision-based dataset that contains 2K image sequences aligned with 12K human-written stories in English.[2] Each image corresponds to a part of a story. Instances of each protagonist are annotated with the character's name (see Figure 1).

We follow Hong et al. (2023b) to use the default data split, that is 11778 for train, 849 for validation,

[2] https://vwprompt.github.io/

and 586 for test[3].

VIST-Character by Liu and Keller (2023) which has visual and textual annotations for recurring characters in 770 stories from the test split of the VIST dataset (Huang et al., 2016), along with an importance rating of all characters in any story.[4] We only use it for evaluation.

We also evaluate on these datasets:

Travel blogs (TB; Park and Kim, 2015) are two datasets with 10K image sequence-story pairs extracted from travel blogs of visiting New York City or Disneyland.

Movie Synopses Associations (MSA; Xiong et al., 2019) contains movie synopses from 327 movies where there are 4494 scenes aligned with corresponding paragraphs in synopses.

2.2 Tracks

We ran two evaluation tracks for this task:

Closed Track focuses on exploring Language and Vision Mapping methods and Language Generation models through a controlled experiment where the visual encoder is fixed. We provide extracted visual features from a pre-trained vision model. Participants must use these features as input (instead of raw images) to train their models on the provided dataset.

Open Track aims to test the state-of-the-art on the task. Participants can use all kinds of resources, including pre-trained models and additional text or vision-only datasets. However, they cannot use other vision and language datasets apart from the provided dataset.

3 Evaluation and Results

In this section, we describe our designs for both automatic and human evaluations for the submissions. The scripts for all automatic metrics be provided after the submission system is open; human evaluation be conducted after all submissions have been received. We release the annotator instructions and source code of all metrics after the shared task.

3.1 Automatic Evaluation

We use metrics in the following categories to evaluate the submissions:

Reference-based metrics including unigram (B-1), bigram (B-2), trigram (B-3), and 4-gram (B-4) BLEU scores (B; Papineni et al., 2002), METEOR (M; Banerjee and Lavie, 2005), ROUGE-L (R; Lin, 2004), and CIDEr (C; Vedantam et al., 2015), which were used in the previous visual storytelling shared task (Mitchell et al., 2018). In our initial proposal, we planned to use BERTScore (BS; Zhang et al., 2020) which is effective in text summarization. Unfortunately, we did not have enough resources to run it by ourselves, because it requires usage of a large amount of GPU time.

Grounding To measure the correctness of referring expressions of human characters in stories, we use the character-matching (CM) metric defined in (Hong et al., 2023a).

Diversity We use metrics used by Hong et al., 2023b including the unique number of verbs, verb-vocabulary ratio, verb-token ratio, percentage of diverse verbs not in the top-5 most frequent verbs, and unique:total ratios of predicate unigram, bigram, and trigram.

Coherence Following Hong et al., 2023b, we use the generative Entity Grid model to calculate the log-likelihood based on entity transitions in system outputs.

3.2 Human Evaluation

In natural language generation tasks, automatic metrics do not provide a full understanding of the quality of the generated text. Reference-based metrics, in particular, have been shown to not correlate well with human judgment. In addition, several important aspects of narratives such as creativity and logical coherence are hard to judge using automatic evaluation. Therefore, we also conducted a human evaluation for the submissions, focused on narrativity (whether the generation is a story or simply a description of images), character grounding (correctness of referring expressions, model hallucinations), and coherence. The scale of the evaluation depends on the funding we have. We also encouraged participants to perform their own human evaluation and include the results in their reports.

3.3 Baselines

We employ two models as baselines for each track.

EntityGrid (Hong et al., 2023b) is the baseline for Closed track. It is a Transformer-based model that adapts the visual features with pre-trained GPT-2.

[3] Please contact the authors for details on the other datasets and how they are applied during the evaluation.

[4] https://github.com/iz2late/VIST-Character

Team	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
baseline	EntityGrid	37.12	13.86	7.33	3.96	34.27	14.78	0.65
team-DMG	LLaVA-S	35.03	14.08	7.90	4.07	34.02	12.16	0.88

Table 1: Performance comparison of different teams on Closed track. All numbers are the higher the better.

Team	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
baseline	GPT-4o	20.71	1.52	0.07	0.00	14.21	10.88	1.21
HTWK	GPT4-RA	19.39	1.47	0.03	0.00	12.53	10.70	0.92
team-DMG	LLaVA-O	22.28	2.56	0.14	0.00	18.09	13.51	1.64

Table 2: Performance comparison of different teams on Open track. †We observed extremely low numbers on BLEU-4. All numbers are the higher the better.

GPT-4-GPT-4o (OpenAI, 2023) is the baseline for Open track.

3.4 Teams and Models

There are two teams that participated in our tasks. One team participated in the Open track only, and the other team participated in both.

HTWK is a team from Leipzig University of Applied Sciences, Germany. They only participated in the Open track. They employ two similarity retrievers to find semantically closest samples from the training set, which serve as examples for the multimodal generative model. First, an image similarity retriever identifies the most similar images for each image in the input sequence. A prompt is then constructed using the retrieved images along with their descriptions, which are provided as examples for the model to generate descriptions for each image. Next, the method concatenates all the generated descriptions and uses a textual similarity retriever to find the most semantically related story. This story serves as the example in the prompt, guiding the model to generate a coherent and reasonable narrative for the input sequence of images.

team-DMG is a team from the University of Amsterdam, Netherlands. They participated in both tracks. For the Closed track, they proposed an updated version of the TAPM model (**LLaVA-S**). To enhance TAPM's performance while maintaining a lower parameter count, they replaced the original language model with LLaVA, a state-of-the-art large language model, and adapted the visual encoder accordingly. They utilized a 4-bit quantized version of LLaVA and fine-tuned it using the LoRA approach, focusing on the multi-head self-attention blocks. Additionally, they improved the vision component by supplementing ResNet-101 features with representations extracted from a pre-trained Vision Transformer (ViTbase) model.

For the Open track, they use a fine-tuned LLaVA model (**LLaVA-O**), which is a general-purpose multimodal foundation model similar to BLIP-2. However, instead of focusing on model architecture, LLaVA emphasizes training data and procedure. It is notable for extending instruction-tuning to the language-image multimodal space by training on vision-language instruction-following data. This data is constructed by querying GPT-4 with various in-context-learning prompts to generate <image, caption> pairs from existing datasets like COCO. LLaVA connects visual features with language embeddings using a single linear layer, unlike BLIP-2, which uses Q-Former. The team uses LLaVA to generate stories in a zero-shot manner under different linguistic context settings.

3.5 Automatic Metrics

Here we summarize the results for the Closed and Open tracks in the tables above.

In the Closed track (Table 1), team-DMG's LLaVA-S model outperforms the baseline EntityGrid model in terms of CIDEr (0.88 vs. 0.65) and BLEU scores, with notable improvements in BLEU-2, BLEU-3, and BLEU-4, although both models perform similarly in METEOR and ROUGE-L. While team-DMG's submission shows competitive performance, the overall improvement in BLEU and CIDEr suggests that the submissions are gradually advancing beyond the baseline's entity-based approach.

For the Open track (Table 2), team-DMG's LLaVA-O model also surpasses the baseline GPT-4o model, achieving the highest scores in BLEU-1, BLEU-2, and ROUGE-L, as well as a significantly better CIDEr score (1.64 vs. 1.21). In comparison, HTWK's GPT4-RA performs slightly lower

	Model	Coherence	Diversity	Grammaticality	Grounding	Overall
Closed	baseline (EntityGrid)	1.53	2.30	3.13	2.17	1.47
	team-DMG (LLaVA-S)	1.72	2.85	2.98	1.74	1.47
Open	baseline (GPT-4o)	4.35	3.76	4.90	4.31	3.65
	HTWK (GPT4-RA)	4.04	3.65	4.94	3.94	3.29
	team-DMG (LLaVA-O)	1.41	2.67	3.08	1.47	1.33

Table 3: Human evaluation of teams in both tracks. Higher numbers are better for all measures.

across all metrics, trailing behind both the baseline and team-DMG in key metrics such as METEOR and CIDEr. Notably, despite the improvements, all systems in both tracks still perform poorly in higher-level BLEU metrics (BLEU-3, BLEU-4), indicating challenges in producing more refined n-gram matches.

Overall, while team-DMG's models consistently improve over the baselines in both tracks, there remains room for further advancements, particularly in terms of the more nuanced and detailed metrics like higher-order BLEU scores. Additional analysis may be needed to explore why these improvements are not more pronounced across all metrics.

3.6 Human Evaluations

The human evaluation results for both the Closed and Open tracks reveal notable differences in system performance across various metrics, including Coherence, Diversity, Grammaticality, Grounding, and Overall scores.

Table 3 presents a comparison of different models evaluated on several criteria under both Closed and Open settings. For the Closed setting, team-DMG (LLaVA-S) achieves a slight improvement in terms of Coherence (1.72) and Diversity (2.85) compared to the baseline (EntityGrid), although both models achieve the same overall score (1.47). Grounding scores remain relatively low for both models in this setting, with team-DMG scoring 1.74 and EntityGrid slightly higher at 2.17.

In the Open setting, the baseline model (GPT-4o) outperforms all other models in nearly every category, with a Coherence score of 4.35, Grammaticality of 4.90, and Grounding of 4.31. HTWK (GPT4-RA) follows closely with slightly lower Coherence (4.04) and Grounding (3.94), but surpasses GPT-4o in Grammaticality (4.94). In contrast, team-DMG (LLaVA-O) shows lower scores across all metrics, particularly in Coherence (1.41) and Grounding (1.47), resulting in the lowest overall score of 1.33.

These results highlight that while team-DMG demonstrates some advantages in Diversity and Coherence under Closed conditions, the Open setting models show a clear dominance of GPT-4o and GPT4-RA, particularly in Grammaticality and Grounding. Overall, the baseline models perform better in terms of general language quality, while team-DMG struggles to match their performance, especially in the Open setting.

3.7 Case Study

We also conduct case study to inspect the generated stories. The results suggest that models like GPT-4o and GPT4-RA are more adept at balancing narrative coherence, character interaction, and environmental immersion, making them suitable for tasks that require rich storytelling and visual grounding. GPT-4o generates stories that are visually grounded on characters while GPT4-RA generates stories that are more grounded on environments. In contrast, models like the baseline EntityGrid and team-DMG LLaVA-O face challenges with redundancy, coherence, and character development. Future improvements could focus on refining character relations and enhancing logical event progression to further bridge the gap between generated narratives and human storytelling.

4 Related Work

VLG with Coherence. One relevant VLG task that requires some sort of coherence in the generated text is movie description (Rohrbach et al., 2015), where the input is a video clip from the movie and the output is the corresponding text description of the scene. Chandu et al. (2019) propose a dataset of procedural text from recipes with instructional images, but characters are not explicitly annotated. Unfortunately, the local coherence of the generated text is not evaluated in either of these tasks (Mitchell et al., 2018). Visual Storytelling is a task that requires local coherence, where the input is a sequence of images and the output is a coherent story (Huang et al., 2016). However, it is not specif-

		R-L	C	D	G	V	O
baseline-EntityGrid	Liam was a soldier that was being held by a soldier . the enemy soldiers were shooting at the enemy soldiers . the soldiers are on top of a village trying to find the enemy . the enemy soldiers were hiding in the middle of the area trying to get them to get out . Oliver , Elijah and James are trying to get the best way to get the next move .	26.53	1.0	2.5	3	2.5	1
team-DMG-LLaVA-S	Liam was checking that the gunfire was over . one soldier started fired shots at the other . he knew they were in trouble . suddenly there was an explosion and people started screaming . he knew it was the right time to escape .	24.54	4.5	4.5	3.5	4	3.5
baseline-GPT-4o	The air was thick with smoke and the acrid scent of gunpowder as Sergeant Jameson led his squad through the war-torn streets. The once bustling city was now a desolate battlefield, with ruined structures and debris marking the path of destruction. Jameson, his helmet slightly askew, carried a rifle slung over his back, his eyes scanning the horizon for any sign of enemy movement. Behind a makeshift barricade of sandbags, Corporal Davis and Private Miller crouched, their faces smeared with dirt and sweat. Davis, gripping a rocket launcher, took aim at a distant tank that rumbled ominously through the haze. Miller, his rifle at the ready, kept a vigilant watch, his heart pounding in his chest. …	12.01	4.0	4.5	5	4.5	4
HTWK-GPT4-RA	Amidst the chaos of a war-torn city, soldiers in combat gear surveyed the battlefield, shrouded in smoke and debris from recent explosions. Taking cover behind sandbags, they prepared for the next wave of combat, their eyes scanning the ruins for any signs of movement. The air was thick with tension as they navigated through the rubble, dodging flames and damaged military vehicles. Explosions echoed in the distance, urging them to stay alert. With every step, they braced for the unpredictable nature of battle, determined to survive the destruction that surrounded them and protect their comrades in this relentless conflict.	13.62	4.5	4.5	5	4.5	4.0
team-DMG-LLaVA-O	Liam and Noah are getting ready to leap a large helicopter from a field that is not already in the air . the soldiers were all aiming to the direction of Olivia . the soldiers were ready to land and go into battle . Noah and Liam look out over the soldiers who are dead . they have fighting to do however and do not have time to go and see him .	16.17	1.5	3	3.5	1.5	2.0

Figure 2: Case study of generated/human-written stories given an example image sequence. We present the ROUGE-L (R-L) for automatic evaluation. We also report the human evaluation scores of Coherence (C), Diversity (D), Grammaticality (G), Visual Grounding (V), and Overall (O) quality.

ically designed to ensure that there are a tellable event sequences depicted in the image sequences, because the dataset used in this task, VIST, is based on photo albums.

Visual Story Generation. Most of the previous tasks for visual story generation have major limitations: there is no sequence of events behind the images (Park and Kim, 2015; Huang et al., 2016) or the dataset is limited in scale (Xiong et al., 2019). None of them can be used for evaluating visual grounding. Mitchell et al. (2018) hosted the first shared task of visual story generation. But there are no automatic evaluations of either coherence or visual grounding. Our shared task is the first to jointly evaluate the coherence and visual grounding of generated stories.

Story Generation There are several existing datasets for generating a story conditioned on a prompt such as previous context (Mostafazadeh et al., 2016), title (Fan et al., 2018), keyword (Yao et al., 2019), cue phrase (Xu et al., 2020), script (Pu et al., 2022), story plot (Rashkin et al., 2020), or detailed plots (Akoury et al., 2020). However, all these datasets relying on textual prompts suffer from the grounding problem that the meanings of textual stories are grounded on textual symbols (Harnad, 1990).

5 Conclusions

We organized the Visually Grounded Story Generation task (VGSG) this year for the first time. Although Visual Language Models have made huge progress in the past couple of years, they are generally not specifically designed with the intention of producing narratively coherent and grounded stories. This task provided further impetus for development in this area. We obtained a couple of submissions, although private communications suggested that other potential participants instead decided to retain their results for publication in other venues. The training data and test platform were mounted on the web and on HuggingFace in order to enable further progress.

Consider the automatic measures: in the controlled (Closed) experiment, the use of a LLaVA model produced what appear to be modest improvements, although the significance of the result is not clear. However, in the Open experiment, it turned out to be difficult to beat GPT-4o with another GPT model, but using a LLaVA-based language model brought noticeable improvements.

However, it should be noted that automatic evaluations do not track the human evaluations. The The submission made no difference in the Closed track to the overall human evaluation, and the best system in automatic evaluation had the worst outcome in human evaluation. This was reflected in our case study. This mismatch between automatic and human evaluation highlights the need for better automatic measures and for future work on this topic to "go the extra mile" and produce robust human evaluations.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. Storium: A dataset and evaluation platform for machine-in-the-loop story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Khyathi Chandu, Eric Nyberg, and Alan W Black. 2019. Storyboarding of recipes: Grounded contextual generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046, Florence, Italy. Association for Computational Linguistics.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Xudong Hong, Vera Demberg, Asad Sayeed, Qiankun Zheng, and Bernt Schiele. 2023a. Visual coherence loss for coherent and visually grounded story generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Xudong Hong, Asad Sayeed, Khushboo Mehra, Vera Demberg, and Bernt Schiele. 2023b. Visual Writing Prompts: Character-Grounded Story Generation with Curated Image Sequences. *Transactions of the Association for Computational Linguistics*, 11:565–581.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–325.

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.

Danyang Liu and Frank Keller. 2023. Detecting and grounding important characters in visual stories. In *37th AAAI Conference on Artificial Intelligence*. AAAI Press.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Margaret Mitchell, Ting-Hao 'Kenneth' Huang, Francis Ferraro, and Ishan Misra, editors. 2018. *Proceedings of the First Workshop on Storytelling*. Association for Computational Linguistics, New Orleans, Louisiana.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. 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.

Cesc C Park and Gunhee Kim. 2015. Expressing an image stream with a sequence of natural sentences. *Advances in neural information processing systems*, 28:73–81.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer.

Dongqi Pu, Xudong Hong, Pin-Jie Lin, Ernie Chang, and Vera Demberg. 2022. Two-stage movie script summarization: An efficient method for low-resource long document summarization. In *Proceedings of The Workshop on Automatic Summarization for Creative Writing*, pages 57–66, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.

Paul Voigtlaender, Soravit Changpinyo, Jordi Pont-Tuset, Radu Soricut, and Vittorio Ferrari. 2023. Connecting vision and language with video localized narratives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2461–2471.

Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. 2019. A graph-based framework to bridge movies and synopses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4592–4601.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. 2022. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Overview of Long Story Generation Challenge (LSGC) at INLG 2024

Aleksandr Migal[♠♣], **Daria Seredina**[♠], **Ludmila Telnina**[♠], **Nikita Nazarov**[♠],
Anastasia Kolmogorova[♠], **Nikolay Mikhaylovskiy**[◊♣]

[♠]National Research University Higher School of Economics,
Saint-Petersburg, Russia, 190068

[◊]Higher IT School, Tomsk State University, Tomsk, Russia, 634050

[♣]NTR Labs, Moscow, Russia, 129594

amigal@ntr.ai, nickm@ntrlab.com

Abstract

This report describes the setup and results of the shared task of human-like long story generation, the LSG Challenge, which asks to generate a consistent, human-like long story (a Harry Potter fanfic in English for a general audience) given a prompt of about 1,000 tokens. We evaluated the submissions using both automated metrics and human evaluation protocols. The automated metrics, including the GAPELMAPER score, assessed the structuredness of the generated texts, while human annotators rated stories on dimensions such as relevance, consistency, fluency, and coherence. Additionally, annotators evaluated the models' understanding of abstract concepts, causality, the logical order of events, and the avoidance of repeated plot elements. The results highlight the current strengths and limitations of state-of-the-art models in long-form story generation, with key challenges emerging in maintaining coherence over extended narratives and handling complex story dynamics. Our analysis provides insights into future directions for improving long story generation systems.

1 Introduction

This report presents an analysis of the results of the Long Story Generation Challenge (LSGC), where participants showcased their systems for creating extended stories. With this shared task, we aimed to advance the generation of long-form literary texts. Our evaluation was based on two main criteria: statistical metrics and a human evaluation protocol. The LSGC was originally proposed by Mikhaylovskiy (2023); this report follows the cited work closely.

Over 110 years ago, mathematician Andrei Markov demonstrated how to study effectively the text using mathematical methods (Markov, 1913). In his work, he examined the relationship between vowels and consonants in the early chapters of Eugene Onegin. He later gave his name to processes known as Markov chains. Markov chains formed the basis of early text generation algorithms that generated basically nonsense based on the probabilistic distribution of words in a text.

Today, text generation has advanced tremendously. Autoregressive probabilistic large language models (LLMs) have become a cornerstone for solving every task in computational linguistics through few-shot learning (Brown et al., 2020) or prompt engineering (Sanh et al., 2021). Many users now interact with advanced commercial models such as GPT, Claude, or Google Bard in chat setting regularly. However, these models still have many deficiencies. Despite the targeted effort, they can generate false information, propagate social stereotypes, and produce toxic language (Taori et al., 2023).

Specifically, current autoregressive language models fail to catch long-range dependencies in the text consistency. While the autoregressive window for commercial models reaches tens or even hundreds of thousands of tokens at the time of writing, which is a lot, it, however, does not allow them to generate long coherent texts. While relevance, consistency, fluency and coherence are relatively easily achieved by the latest

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 47–53
September 23 –27, 2024.

autoregressive generative models on short texts (under 10K tokens), all the current models fail when one tries to generate a long story in a single pass. Modeling long stories requires many additional abilities compared to short texts (Guan et al., 2022), including (1) commonsense reasoning regarding characters' reaction and intention, and knowledge about physical objects (e.g., "river") and abstract concepts (e.g., "irony"); (2) modeling discourse-level features such as inter-sentence relations (e.g., causality) and global discourse structures (e.g., the order of events); and (3) the generation coherence and controllability, which require both maintaining a coherent plot and adhering to controllable attributes (e.g., topics).

Several authors have shown theoretically and empirically (Lin and Tegmark, 2017, Alvarez-Lacalle et al., 2006, Mikhaylovskiy and Churilov, 2023) that the power law autocorrelations decay is closely connected to the hierarchical structure of texts. Indeed, the hierarchical structure of, for example, Leo Tolstoy's War and Peace consists of at least 7 levels: the whole novel, books, parts, chapters, paragraphs, words, and letters. There are strong reasons to think that this structure reflects an important aspect of human thinking: people do not generate texts autoregressively. Writing a long text requires some thinking ahead, and going back to edit previous parts for consistency. This going back and forth can be reflected by navigating a tree-like structure. The autoregressive nature of the current state-of-the-art models does not reflect this; for example, S4 model (Gu et al., 2021) exhibits clear exponential autocorrelations decay (Mikhaylovskiy and Churilov, 2023).

2 Task Description

The LSG Challenge task required participants to provide a system that could output a coherent, human-like long story (a Harry Potter fanfiction for a general audience of at least 40,000 words) given a prompt of about 1,000 tokens. The organizers provided a set of story starters for developers. Systems were evaluated based on text generated from these starters, written by volunteers and imitating the stylistic features of Harry Potter fan fiction. The starters were designed from scratch specifically for this task.

It is important to note that no copyrighted texts were used in the creation of our dataset. The evaluation protocol below also does not require the usage of any of the original Harry Potter texts. It is based on the assumption that the assessors have a general knowledge of the Harry Potter universe, and this is enough to rate the texts using the provided questionnaries.

We employ both automatic and human evaluation to evaluate the quality of the texts. In particular, we used GAPELMAPER (Mikhaylovskiy, 2023) as an unreferenced automatic, statistical metric of the text structuredness. We adopt multiple human evaluation metrics to better measure model performance. Similarly to Kryscinski et al. (2019), we ask annotators to rate the texts across four dimensions:

1. Relevance (of topics in the text to the expected ones),
2. Consistency (alignment between the parts of the text),
3. Fluency (quality of individual sentences)
4. Coherence (quality of sequence of sentences).

Extending Guan and Huang (2020) we ask annotators to rate repeating similar texts. Finally, we asked the annotators to evaluate the creative dimensions of the resulting texts:

5. Doubt of the characters of the text or the narrator in their own rightfulness
6. Expression of the strong positions of the text (beginning/end of the text, beginning/end of the chapter)
7. General idea of the text
8. Usage of idioms
9. Creativity of the text
10. Emotionality of the text

3 Dataset Description

Story starters were created by undergraduate students majoring in Linguistics as a part of their coursework with a proper credit. For testing and development purposes, we presented participants with five distinct story starters.

4 Shared Task Timeline

The LSGC was planned throughout the recent academic year. The key dates of the shared task were:

- SEP, 2023: The shared task is announced at the INLG 2023 conference.
- DEC, 2023: The task website is up; participants can register to the task.
- JULY 15, 2024: The submission is closed; organizers conduct manual evaluation.
- AUG, 2024: The LSG Challenge shared task is fully completed. Organizers submit participant reports and challenge reports to INLG 2024.

5 Baseline

We developed a baseline, published at https://lsgc.vercel.app/baseline, that generates a fan fiction text complying to the shared task requirements to make sure that the shared task is feasible. In light of the shared task's objective to create a lengthy, coherent fanfiction, we incorporated a hierarchical prompting system into the baseline to ensure the narrative's "completeness".

The baseline implements a process that begins with a "story starter". By establishing a clear narrative structure, we create a framework for generating additional content, with the aim of remaining faithful to the original story in the fanfiction we produce. After setting up the narrative framework, we then focus on fleshing out the details of each section, creating chapter outlines that outline the events to be included (see Figure 1). The number of chapters produced will depend on the capabilities of the generative model to generate believable text. This includes:

- Introduction: Establishing the protagonist's world and introducing key themes.
- Development: Presenting obstacles, conflicts, and character growth.
- Climax: A turning point where the protagonist faces a critical challenge or revelation.

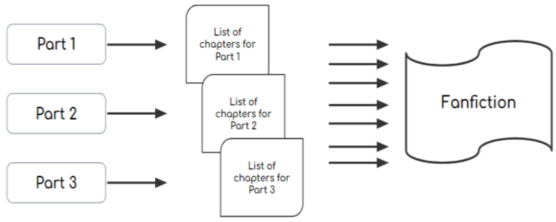


Figure 1: Chapter development

- Resolution: Tying up loose ends, providing closure and a sense of accomplishment.
- Conclusion: Offering a satisfying denouement, wrapping up the narrative and leaving a lasting impression on the reader.

6 Participants

Two teams participated in the challenge. Each team submitted one story generated using their systems. All texts were anonymized prior human evaluation to ensure objective evaluation. Each text was assessed using the GAPELMAPER metric and the human evaluation described below.

Team 1 (Decision Stump, Boriskin, Galimzianova, 2024) – The approach does not include any fine-tuning and utilizes Llama 3 with 70b parameters with special prompting scheme for the text generation. Team 1 developed the baseline in the direction of generating of the book components. The full pipeline consists of 2 parts – summary generation and generation of chapters in a loop with the transmission of context about previous events in the book via the system prompt.

The team presented a text consisting of 14 chapters, each chapter spanning 10 pages. This design aims to have only 14 potential points of discontinuity (at the junctions between chapters) where plot inconsistencies might arise, such as repeated scenes. For instance, at the end of Chapter 1, the main character Theo encounters the heroine Pansy, and at the beginning of Chapter 2, the model again describes their meeting. However, even such minor flaws blend reasonably harmoniously into the overall context. Throughout the fanfic, the narrative thread is maintained, making it

Team	Power law MAPE	Exp law MAPE	GAPELMAPER
Team 1 (Decision Stump)	0.52	0.57	0.91
Team 2 (Neurowling)	0.17	0.40	0.44
Baseline	0.17	0.31	0.57

Table 1: GAPELMAPER metrics of solutions

challenging to distinguish the text from that of a real author.

Team 2 (Neurowling, Seredina, 2024) – Approach is based on fine-tuning the Mistral-7B-Instruct-v0.2-GPTQ model with Supervised Learning (SL). The final text of a fanfiction was generated with the fine-tuned model and the prompts following the baseline.

The team also delivered commendable results: their text comprises numerous short chapters with rapidly unfolding action, unlike the first team's story. This format makes for easier reading, but due to the brevity of each chapter (approximately one page), inconsistencies and contextual discrepancies can be noticed at the chapter boundaries. For example, a character's gender might be female in one chapter and male in another. Nevertheless, the provided structure, which involved preliminary generation of all chapters according to a unified concept (outline), ensured reasonable consistency of the narrative.

7 Evaluation

7.1 GAPELMAPER Metric

GAPELMAPER (GloVe Autocorrelations Power/Exponential Law Mean Absolute Percentage Error Ratio) is a metric designed to assess text coherence based on the autocorrelation of embeddings. It helps determine whether the text is intrinsically structured, based on the decay patterns of the autocorrelations. The results of evaluating the submitted texts with GAPELMAPER are listed in the Table 1.

Mikhaylovskiy and Churilov (2023) state that “GAPELMAPER less than 1 means that the autocorrelations decay according to a power law and the text is structured in a way. GAPELMAPER more than 1 means that the autocorrelations decay according to an exponential law and the text is unstructured”. From this viewpoint, the text produced by the system by Decision Stump is on a verge of being structured, while Neurowling’s text exhibits a clear long-distance structure to a level that exceeds the baseline.

7.2 Human Assessment

To assess the results of our shared task from a human perspective, we asked a group of undergraduate students majoring in Linguistics to read several fanfics about "Harry Potter", including texts written by humans and those generated by language models participating in our shared task. The average age of the evaluators is 21 years old; all of them are confident English speakers (B2 to C1 level as assessed via prior coursework). The native language of all evaluators is Russian. Some respondents had only read "Harry Potter" in Russian and have never read any "Harry Potter" books in English and were therefore surprised by the absence of explanations and hints about the characters' backstories, with many terms, such as "Sorting Hat", being unfamiliar to them. This lack of context sometimes led to difficulties in understanding the narrative and its underlying themes.

Each evaluator evaluated three texts, randomly selected between participant submissions, baseline and three human-written fan fictions. The number of persons who evaluated the work of the Decision-Stump team was 10, while only 5 persons evaluated the text of the Neurowling team. The respondents were asked to answer a series of questions about the texts they read (the results can be seen in the table) and provide any additional comments they might have.

The evaluators analyzed the texts for literary quality, originality, style, cohesion and coherence of the generated texts and overall perception. Each evaluator assessed the text according to the documented criteria on a scale from 1 to 5, where 1 is the worst rating possible and 5 is the highest. The tables 2 and 3 show the average scores of the calculated based on all expert assessments of the data. We present the results of “Harry Potter and the Slytherin Selection” (DrizzleWizzle, 2012) evaluation as “Fan Fiction” line for comparison.

	Relevance		Consistency	The order of events	Repeating similar texts	Fluency	Coherence
Team	Correlation between the fanfic title and its content	Compatibility of chapter and sub-chapter titles with the overall style of the text	The strength of the stylistic connection between all the elements of the text	The pace of the plot	Word repetitions	Text composition	Text syntax
Decision Stump	1.75	**3.6**	2	2.3	2.8	2.3	3
Neuro-wling	**3.25**	3.2	2.6	1.8	2.6	2.8	3.2
Baseline	2.25	3.5	3	2.8	3.6	3.2	3.8
Fan Fiction	2.3	2.7	**3.9**	**3**	**3.9**	**4.3**	**3.9**

Table 2: Human evaluation results – text quality

Text 1 ("Decision Stump")

The majority of respondents noted the presence of narrative inconsistencies in the texts, stating that "instances of redundancy occur not only on a lexical level but also on a semantic level: the same event can be described multiple times using slightly different words or with different (not very original) details, which may be indicative of a lack of cohesive narrative structure". Additionally, respondents pointed out the lack of character dialogue in the texts, which made the stories seem less engaging: "The story is driven not by the characters and their actions, but by the narrative itself, resulting in a sense of detachment from the characters' experiences". Respondents who had read the original books in English or were fans of the series noted stylistic discrepancies: "There are moments that stand out to a reader immersed in the lore, indicating that the text was clearly not written by an expert (for example, the way Hagrid speaks, which deviates from his characteristic mannerisms and speech patterns in the original books)".

Nevertheless, many respondents noted that the text has some strong aspects, such as a well-structured beginning and conclusion, and a moderate use of complex syntactic structures (embedded clauses, subordinate clauses of various types, participial phrases, impersonal or indefinite-personal sentences, and ellipses). The text employs conventional stylistic devices, but the language itself is not sufficiently creative. Respondents were unable to discern the main and overarching idea or theme in the text, although occasional glimpses of an idea did emerge in certain sections. Furthermore, the text also contains elements that appear to be logically integrated into the narrative, but ultimately prove to be inconsequential to the overall plot. These elements seem to be introduced with a specific purpose in mind, but fail to contribute meaningfully to the story's development or resolution, leaving the reader wondering about their significance. On the other hand, respondents praised the harmonious combination of chapter and subchapter titles with the overall style of the text.

Text 2 ("Neurowling")

The informants highly praised the semantic correspondence between the fanfiction title and the subsequent text, as well as the combination of chapter and subchapter titles with the overall style and content of the chapters and subchapters. They noted the presence of hints at a common idea or theme, although it was challenging to pinpoint a single, unified concept. However, the informants were less impressed with the pacing of the plot, which they found to be either too fast or too slow at times, with the rhythm sometimes changing in a way that didn't align with the unfolding narrative.

On a lexical and grammatical level, the text exhibited repetition, which came across as a limited vocabulary. Nevertheless, the text featured a

Team	Doubt of the characters of the text or the narrator in their own rightness	Expression of the strong positions of the text (beginning/end of the text, beginning/ end of the chapter)	General idea of the text	Usage of idioms	Creativity of the text	Emotionality of the text
Decision Stump	2.1	2.9	2.1	2.4	2	2.6
Neuro-wling	2.4	3.2	3.2	3	3	3
Baseline	3.1	3.2	3.2	3.1	3.5	3.6
Fan Fiction	3.5	3.4	4	4.2	3.3	4.4

Table 3: Human evaluation results – creative aspects

sufficient number of complex constructions, including parenthetical phrases, subordinate clauses, and participial phrases. The text also employed conventional metaphors, comparisons, and familiar clichéd oxymorons, but nothing beyond that.

When asked if they could summarize the main plot of the text, some informants responded positively, which suggests that there are indeed signs of a cohesive narrative. The majority of informants also praised the strong opening and conclusion of the text. Regarding the emotional resonance of the text, this aspect of literary writing still leaves room for improvement, as the emotions expressed in the text change, but in a somewhat abrupt and peculiar manner.

Furthermore, the chapters often repeated each other's plot, which led one informant to comment, "This makes me think that it wasn't written by a human. If it weren't for this, I would say that the text was written by a teenager who is very fond of the Harry Potter universe."

8 Conclusions

Both teams have demonstrated their capacity to generate long-form narratives with structured coherence, as evidenced by their GAPELMAPER scores. However, based on the combined quantitative and qualitative evaluations, Team 2 ("Neurowling") emerges as the stronger contender. Both teams not very significantly departed from the baselines in terms of the system architecture. The results of both teams also only sparsely improved on the baseline.

The GAPELMAPER score of 0.44 for Team 2 indicates a significantly more cohesive narrative structure compared to Team 1's score of 0.91. Although both texts exhibited certain narrative inconsistencies, Team 2's shorter chapter format and faster pacing made the text more accessible to readers, even if this format occasionally led to contextual discrepancies. Moreover, the manual evaluation highlighted that Team 2's text maintained a better alignment between chapter titles and content, as well as a clearer thematic structure.

While Team 1 ("Decision Stump") produced a more extensive narrative, the manual assessment revealed that this length led to redundancy and a lack of emotional engagement, as well as difficulties for readers unfamiliar with the "Harry Potter" universe. In contrast, Team 2's text, despite its flaws, was more favorably received in terms of readability and structure.

The evaluators easily detect the generated texts. The generated texts are still behind even non-professionally writing humans in terms of text quality and creativity.

Nevertheless, we can say that our expectations for this challenge were reasonably justified. The results of this study show the difference between using fine-tuning and prompt engineering approaches in text generation and demonstrate the advantages and disadvantages of each. In future, we would like to continue this research with a larger data set, and see more diverse text generation approaches from participants. This would allow us to get closer to understanding the linguistic nature of the generated text and, possibly, the nature of the text itself.

References

Enric Alvarez-Lacalle, Beate Dorow, Jean-Pierre Eckmann, and Elisha Moses. 2006. *Hierarchical structures induce long-range dynamical correlations in written texts*. PNAS, 103(21):7956–7961.

Aleksandr Boriskin, Daria Galimzianova, 2024. *The LSG ChallengeWorkshop at INLG 2024: Prompting Techniques for Crafting Extended Narratives with LLMs*. In Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, Tokyo, Japan. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, et al. 2020. *Language models are few-shot learners.* In Advances in Neural Information Processing Systems, volumes 2020-Decem, pages 1877–1901.

DrizzleWizzle. 2012. *Harry Potter and the Slytherin Selection*. Retrieved from: https://www.fanfiction.net/s/8666085/1/Harry-Potter-and-the-Slytherin-Selection

Albert Gu, Karan Goel, and Christopher Ré. 2021. *Efficiently Modeling Long Sequences with Structured State Spaces.* International Conference on Learning Representations. 2021:1–32.

Jian Guan and Minlie Huang. 2020. *UNION: An Unreferenced Metric for Evaluating Open-ended Story Generation.* In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9157–9166, Online. Association for Computational Linguistics.

Jian Guan, Zhuoer Feng, Yamei Chen, Ruilin He, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2022. *LOT: A Story-Centric Benchmark for Evaluating Chinese Long Text Understanding and Generation.* Transactions of the Association for Computational Linguistics, 10:434–451.

Henry W. Lin and Max Tegmark. 2017. *Critical behavior in physics and probabilistic formal languages.* Entropy, 19(7):1–25.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. *Neural text summarization: A critical evaluation.* 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Andrei Markov, 1913. *An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains.* Science in Context. 2006. Vol. 19, no. 4. pages 591–600. DOI 10.1017/S0269889706001074.

Nikolay Mikhaylovskiy. 2023. *Long Story Generation Challenge*. In Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges, pages 10–16, Prague, Czechia. Association for Computational Linguistics.

Nikolay Mikhaylovskiy and Ilya Churilov. 2023. *Autocorrelations Decay in Texts and Applicability Limits of Language Models*. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2023"

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, et al. 2021. *Multitask Prompted Training Enables Zero-Shot Task Generalization.* ICLR 2022.

Daria Seredina, 2024. *A Report on LSG 2024: LLM Fine-Tuning for Fictional Stories Generation.* In Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, Tokyo, Japan. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. *Stanford Alpaca: An Instruction-following LLaMA Model.* https:github.com/tatsu-lab/stanford_alpaca, 2023.

RDFpyrealb at the GEM'24 Data-to-text Task: Symbolic English Text Generation from RDF Triples

Guy Lapalme
RALI-DIRO / Université de Montréal
CP. 6128, Succ. Centre-Ville
Montréal, Québec, Canada
lapalme@iro.umontreal.ca

Abstract

We present a symbolic system, written in Python, used to participate in the English Data-to-text generation task of the *GEM Shared Task at the Generation Challenges (INLG'24)*. The system runs quickly on a standard laptop, making it fast and predictable. It is also quite easy to adapt to a new domain.

1 Introduction

This paper describes pyrealb, a system for tackling the Data-to-text generation task of the *GEM Shared Task at the Generation Challenges (INLG'24)* (Mille et al., 2024). It uses a symbolic approach to this problem, which has become almost *forgotten* due to the popularity of neural networks and large language models. We thought it would be interesting to compare the results between computationally intensive methods that can sometimes be difficult to control with a *predictable*, lightweight and fast symbolic approach.

The system is conceptually simple, each RDF triple corresponds to a sentence in which the subject and the object of a triple are mapped almost verbatim as subject and object of the sentence. The predicate of the triple corresponds to a verb phrase that determines the structure of the sentence. The system orders predicates to create a meaningful story, and merges parts of sentences when they have shared subjects or predicates. The final realization is performed using pyrealb, a French-English realizer used in some data to text applications (Lapalme, 2023).

pyrealb derived from our submission to the WebNLG Challenge (Lapalme, 2020) 2020 in which the text realization was performed through an API that sent JSON structures to a jsRealB[1] server that returned the final text. In this case, we perform the realization directly in Python. Our paper provided a critical review of the data and discussed the suitability of this competition results in a wider Natural Language Generation setting. These remarks are still valid, given that the data for this shared task is the same or a textual replacement of entities without changing the organization of the RDF triples. Provisions have been made to remove singleton sets from the evaluation in this competition, thus making sentence realization a bit more challenging.

2 Text Generation

We recall that an RDF triple is composed of three URIs. In this dataset, they are replaced by English tokens, corresponding to the subject, the predicate and the object. An object can also be a constant string, a date or a number. The predicate of a triple declares a relation between the subject and the object, such as `Campeonato_Brasileiro_Série_C | country | Brazil`, in which `Campeonato_Brasileiro_Série_C` is the subject (a Brazilian Soccer competition), `country` the predicate indicating that the subject *takes place in the country* indicated by the object `Brazil`.

To illustrate our NLG process, we use the set of triples shown in Table 1 with the corresponding generated English sentence.

The first step in text generation is to determine what information to include in the text. In the context of this shared task, this is given: it consists of at most 7 triples, and only 4% of the sets are made up of seven triples. Moreover, 20% of the triples are singletons that are easy to generate, but they were not submitted for human evaluation. Since the predicate of a triple indicates a relationship between its subject and object, in our case, it maps to a verb that links the subject and object of the sentence realizing this triple.

[1] https://github.com/rali-udem/jsRealB

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 54–58
September 23 –27, 2024.

```
<entry category="SportsTeam" eid="Id649" shape="(X␣(X)␣(X)␣(X␣(X))␣(X␣(X)␣(X)))" shape_type="mixed" size="7"
    >
    <modifiedtripleset>
        <mtriple>Estádio_Municipal_Coaracy_da_Mata_Fonseca | location | Arapiraca</mtriple>
        <mtriple>Agremiacao_Sportiva_Arapiraquense | league | Campeonato_Brasileiro_Série_C</mtriple>
        <mtriple>Campeonato_Brasileiro_Série_C | champions | Vila_Nova_Futebol_Clube</mtriple>
        <mtriple>Campeonato_Brasileiro_Série_C | country | Brazil</mtriple>
        <mtriple>Agremiacao_Sportiva_Arapiraquense | numberOfMembers | 17000</mtriple>
        <mtriple>Agremiacao_Sportiva_Arapiraquense | ground | Estádio_Municipal_Coaracy_da_Mata_Fonseca</
            mtriple>
        <mtriple>Agremiacao_Sportiva_Arapiraquense | manager | Vica</mtriple>
    </modifiedtripleset>
</entry>
```

Agremiacao Sportiva Arapiraquense has Vica as manager, it has 17,000 members and plays in the Campeonato Brasileiro Série C league. It plays in Estádio Municipal Coaracy da Mata Fonseca located inside Arapiraca. Campeonato Brasileiro Série C is from Brazil and where Vila Nova Futebol Clube were champions.

Table 1: The top part shows a triple set from `D2T-1-FA-WebNLG_Factual.xml`, the content of the `originaltripleset` is not shown here because it is ignored in the competition. The bottom part shows the realized sentence produced by RDFPYREALB from this input.

2.1 Microplanning

Since triples are unordered, the first critical step is organizing them to create an *interesting story*. First, we group the triples based on their subjects. We then sort the triples within each group. For example, when describing a person, we can begin with their date and place of birth, then move on to their activities, before finishing with their retirement and death. For a university or a football club, we would start with its creation date, then its directors and finally its activities. To achieve this ordering, each predicate is assigned a *priority* that is used to sort the triples. These priorities were established by hand and are currently independent of the category of the subject.

Then the groups are processed in descending order of triplets. We also query DBpedia to determine whether the category of a group subject corresponds to the specified category in the data. If so, we increase its score so that the text begins with this subject. Each group forms a sentence as a coordination of *subsentences*. Because long coordinated sentences are often difficult to follow, groups of more than three triplets are split into two sentences. In order to avoid very short sentences, a group with a single triple is combined using a subordinate when its subject is the object of another triple in a bigger group. Table 1 shows an example of this in which the last triple of the first group has been combined with the last triple.

Table 2 shows the result of the sorting and grouping process on the example of Table 1. The four triples having `Agremiacao_Sportiva_Arapiraquense` as subject are grouped and sorted to form a coherent story. This input is used for realizing the three sentences shown in the bottom part of Table 1 using PYREALB.

```
Agremiacao_Sportiva_Arapiraquense
    manager Vica;
    numberOfMembers 17000;
    league Campeonato_..._C;
    ground Estádio_..._Fonseca.
Campeonato_..._C
   country Brazil;
   champions Vila_Nova_Futebol_Clube.
Estádio_..._Fonseca
    location Arapiraca.
```

Table 2: `mtriples` from Table 1 sorted and grouped, shown as a Turtle-like formalism, used as input for RDFPYREALB. Predicates and objects sharing the same subject are shown indented and separated by semicolons. Some tokens are shown here with ellipsis to make them fit in the two-column format. The bottom of Table 1 corresponds closely to this *text plan*.

2.2 Surface realization

For the final realization step, we use PYREALB a Python implementation of JSREALB (Lapalme, 2022) in which programming language instructions create data structures corresponding to the constituents of the sentence to be produced. Once the data structure is built, it is traversed to produce the list of words in the sentence, taking care of issues such as conjugation, agreement, capitalization, and other *small* details that help readers and evaluators.

The data structure is built by calls to functions whose names were chosen to be similar to the symbols typically used for constituent syntax trees, such as a *Terminal* (e.g. `N` (Noun), `V` (Verb), `A` (adjective), `D` (determiner), `Q` which quotes its parameter thus allowing *canned text*) or a *Phrase* (e.g. `S` (Sentence), `NP` (Noun Phrase), `VP` (Verb Phrase)).

```
S(Pro("I").g('n'),
  VP(V("play"),
     PP(P("in"),
        SP(Q("Estádio␣Fonseca"),
           VP(V("locate").t('pp'),
              PP(P("inside"),
                 Q("Arapiraca")))))))
```

Table 3: Top: Python functional notation for a PYREALB expression realized as: It plays in Estádio Fonseca located inside Arapiraca

Features added to structures with the dot notation can modify their properties. Terminals can specify their person, number and gender. Phrases can have a negation or be put into passive mode. A noun phrase can be pronominalized, and coordinated phrases are automatically processed, inserting appropriate commas and conjunctions between coordinated elements. Table 3, shows the Python calls to create an internal structure that is realized as an English sentence.

2.3 Sentence Templates

The goal is to transform the structure of Table 2 into that of Table 3. We have manually defined 250 templates corresponding to the most frequent predicates in the set (those with 10 or more occurrences). When no defined template can be found, we use a default template (described in Section 2.4), which was used in 5% of cases.

A predicate `p` corresponds to a Python lambda expression whose parameter is the object `o`. The predicate is called to create a sentence with the subject `s`. The actual parameters are quoted strings of the subject or object of the triple, but replacing underscores by spaces with special cases for numbers and dates.

For example, given the two following Python definitions:

```
managerP = lambda o: VP(V("have"),
                        o,
                        Adv("as"),
                        N("manager"))
sentence = lambda s,p,o: S(Q(s),
                           p(Q(o)))
```

the call

```
sentence("Agremiacao",
         managerP,"Vica")
```

creates the following structure:

```
S(Q("Agremiacao"),
  VP(V("have"),
     Q("Vica"),
     Adv("as"),
     N("manager")))
```

which is verbalized as Agremiacao has Vica as manager. by PYREALB. This is the basic mechanism for creating sentence structures that can be combined in various ways.

```
"city": (30, False, [
    lambda o:VP(V("be"), _from(o)),
    lambda o:VP(_vpas("locate"), _in(o))]),
"country": (40, False, "city"),
"ground": (50, True, [
    lambda o:VP(V("play"), _in(o))]),
"league": (50, True, [
    lambda o:VP(V(oneOf("be", "play", "compete")),
        _in(NP(D("the"), o, N("league"))))]),
"manager": (20, False, [
    lambda o:VP(_vpas("manage"), _by(o)),
    lambda o:VP(V("have"),o,
              Adv("as"),N("manager"))]),
```

Table 4: A few Python templates using auxiliary function to build passive verbs (`_vpas`) or prepositional phrases such as `_from(..)` or `_in(..)`

Templates are organized in a dictionary (see Table 4). The name of the predicate is the key, and the value is a 3-tuple with the following elements: a priority (a number between 0 and 100) used for sorting, a boolean indicating if its subject can be a human, and a list of lambda expressions that can verbalize this predicate, one of which is randomly chosen at the realization time.

Templates associated with predicates were developed by looking at `lex` elements in the original WebNLG training corpus. When two templates have the same realizations, the third element of the pair is the name of the original predicate (see `country` in Table 4).

Once we agreed on this template structure, writing them became relatively easy. It takes less than minute to write a lambda defining a constituent expression to reproduce some of them. We noticed that many lexicalizations are often very similar; crowd workers seem to often rely on copy-pasting the subject and the object.

Unfortunately, the names of the predicates used in the Wikidata dataset were different for the same relation. So we developed a mapping between them, as shown in Table 5.

2.4 Default Template

When a predicate is not in the table, a default template is created. By detecting case changes, the name of the predicate is split into *words* and taken as the subject of the be auxiliary, the object is used as an attribute. For example,

```
servedAsChiefOfTheAstronautOfficeIn =>
  Q("served␣as␣chief␣of␣the␣astronaut␣
     office")
```

In the final sentence, the subject of the triple is taken as subject of the be auxiliary, the object of

```
wikidata_properties = {
    'Occupation':"occupation",
    'PlaceOfBirth':"birthPlace",
    'DateOfBirth':"birthDate",
    'PositionHeld':"position",
    'HasChild':"have_as_child",
    'PlaceOfDeath':"deathPlace",
    'Spouse':"spouse",
    'ParticipantIn':"competeIn",
    'HasFather':"have_as_father",
    ...
}
```

Table 5: Mapping between the names of predicates used in the Wikidata dataset used as key and the name used in the WebNLG dataset. When a name contains an underscore (e.g., `have_as`), then a custom verb phrase pattern is used.

the triple is used as an attribute. For example, the triple

```
Alan_Shepard |
    servedAsChiefOfTheAstronautOfficeIn
    | 1963
```

is realized as Alan Shepard served as chief of the astronaut office in is 1963. which is not colloquial but understandable.

2.5 Text aggregation

In some cases, dealing with related information (e.g., birth date and place), combining templates using only their complements (i.e., their last element) will simplify the text. For this we define groups of predicates that can be combined at realization time. When two or three triples are merged into a single sentence, the subject is used at the start but a pronoun is used for the following references. Currently, a very simple system is used for choosing the pronoun: if the predicate is coded as being applicable to a human and the gender of the subject obtained by querying DBPedia is `male`, he is used, if it is `female` then she is chosen, otherwise it is used. When a single triple whose subject is used as object of another, it is combined with the subordinate using a pronoun: who if the predicate applies to a human, otherwise that.

3 Running the System

The PYREALB is publicly available, its source code[2] is licensed under Apache-2.0 and the linguistic resources are licensed under CC-BY-SA-4.0. It can also be used as a PyPi module.[3]

The Python code for RDFPYREALB is a demo[4] of PYREALB. The demo, launched with `WebGenerate.py`, is illustrated with English and French texts realized from 6 and 7 triples selected from the original WebNLG 2020 data, which give rise to the most interesting and challenging texts. The script for realizing the submissions to this shared task is `GEM-2024.py`.

4 Comments on the task data

#sbj	WN-trn	WN-dev	WN-test	WkData
1	57%	60%	74%	88%
2	33%	31%	20%	9%
3	9%	8%	5%	2%
4	<1%	<1%	-	<1%
#tpl	13 124	1 667	1 779	1 712

Table 6: percentages of the number of subjects in different triple sets (WN- is WebNLG-2020, trn, dev and test). WN-tst WebNLG-based (D2T-1) and WkData is the Wikidata-based (D2T-2) of this competition.

In a previous paper (Lapalme, 2020), we argued that the *simplified triple* format of WebNLG does not adequately represent the problem of realizing semantic web data. It short-circuits many important issues, such as the lexical selection of the subject and object. Additionally, the relation names do not conform to the well-established W3C naming conventions. We now raise another issue that we did not notice at the time: the number of distinct subjects in triple sets. Table 6 shows the distribution of the number of subjects in the *factual* sets of data; since the both *fictional* and the *counterfactual* were derived from the factual, their distribution is the same. We see that the vast majority of triples have a single subject: 74% for WebNLG and 80% for WikiData. This simplifies greatly the problem of the text organization leaving only the problem of splitting into one or two sentences.

5 Conclusion

This paper described a symbolic approach to tackling the GEM 2024 SHARED TASK. The approach relies on PYREALB, an existing text realizer that takes care of most of the low-level grammatical aspects, so the pattern could be specified at a relatively high level. After a few false starts and once the overall program organization was settled, it was relatively easy for me to develop and organize the patterns. The preliminary automated scores seems quite competitive compared to those of the other

[2] `https://github.com/lapalme/pyrealb`
[3] `https://pypi.org/project/pyrealb/`
[4] `https://github.com/lapalme/pyrealb/tree/main/demos/RDFpyrealb`

participants, whom we conjecture mostly used machine learning approaches. In fact, almost all systems seem to obtain quite similar results depending on the scoring method. RDFpyRealB is very fast and can easily be adapted to new domains. Considering that adding one new predicate takes about one minute, developing 250 new ones would take about four hours. Machine learning could be used to develop new templates, although we doubt that it would be any faster.

References

Guy Lapalme. 2020. RDFjsRealB: a symbolic approach for generating text from RDF triples. In *WebNLG 2020 : 3rd Workshop on Natural Language Generation from the Semantic Web*, pages 144–153, Dublin, Ireland (virtual). SIGGEN.

Guy Lapalme. 2022. The jsRealB text realizer: Organization and use cases. (arXiv:2012.15425).

Guy Lapalme. 2023. Data-to-text bilingual generation. Technical Report arXiv:2311.14808.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

DipInfo-UniTo at the GEM'24 Data-to-Text Task: Augmenting LLMs with the Split-Generate-Aggregate Pipeline

Michael Oliverio, Pier Felice Balestrucci, Alessandro Mazzei and Valerio Basile
University of Turin - Italy Computer Science Department
michael.oliverio@edu.unito.it
pierfelice.balestrucci@unito.it
alessandro.mazzei@unito.it
valerio.basile@unito.it

Abstract

This paper describes the DipInfo-UniTo system participating to the GEM Shared Task 2024. We participate only to the Data-to-Text (D2T) task. The DipInfo-UniTo system is based on Mistral (Jiang et al., 2023), a recent Large Language Model (LLM). Most LLMs are capable of generating high-quality text for D2T tasks but, crucially, they often fall short in terms of adequacy, and sometimes exhibit "hallucinations". To mitigate this issue, we have implemented a generation pipeline that combines LLMs with techniques from the traditional Natural Language Generation (NLG) pipeline. In particular, we have a three step process *SGA*, consisting in (1) *S*plitting the original set of triples, (2) *G*enerating verbalizations from the resulting split data units, (3) *A*ggregating the verbalizations produced in the previous step.

1 Introduction

In the last few years, LLMs have become the state of the art in natural language generation tasks, as can be seen in the most important conferences and challenges, such as the INLG conference and the WebNLG challenge.[1][2] LLMs enable high performance across various fields of NLP, including RDF-to-Text. The use of such models can occur through prompting techniques or, if there is a suitable dataset available for their task, by fine-tuning the models, which involves further training. This latter approach leaded to improved performances in many tasks related to generation. Systems like LLaMA2 (Touvron et al., 2023) and Mistral are among the most popular open-weights system for text generation. Fortunately, a linguistic resource for fine-tuning these models is provided by the WebNLG challenge. This corpus, originally designed for English and later extended to other languages (German (Ferreira et al., 2018), Russian (Shimorina et al., 2019) and partially Maltese (Cripwell et al., 2023), among others), consists of data units, i.e., sets of RDF triples, composed of subject, predicate, and object, accompanied by their verbalizations, which represent the semantics of the triples. The system employed in the GEM Shared Task (Mille et al., 2024) consists of a three-step pipeline, which we called SGA (split-generate-aggregate). It includes a Data Unit Splitting Algorithm (S) to simplify data units for subsequent steps, an RDF-to-Text System (G) designed to generate verbalizations from obtained data units, and a Sentence Aggregation System (A) to combine the verbalizations produced in the previous steps.

The paper is structured as follows: in Section 2 we provided a brief selection of related work; in Section 3 we give few details about GEM Shared Task; in Section 4 we give some details on WebNLG 3.0, that is our training corpus; in Section 5 we describe the SGA pipeline; in Section 6 we present the official results of the DipInfo-UniTo system and, finally, Section 7 closes the paper by considering future development. The code and submitted outputs of the DipInfo-UniTo system can be found on GitHub.[3]

2 Related Works

Over the years, RDF-to-Text has become an increasingly important task. Several WebNLG challenges have been held (2017, 2020, and 2023) to develop the best RDF-to-Text models based on WebNLG corpora.[4][5][6] A common strategy involves using prompting techniques or fine-tuning to generate verbalizations from given RDF triples (Wang

[1] https://aclanthology.org/venues/inlg/

[2] https://synalp.gitlabpages.inria.fr/webnlg-challenge/

[3] https://github.com/MichaelOliverio/DipInfo-UniTo-GEM24

[4] https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2017/

[5] https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2020/

[6] https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2023/

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 59–65
September 23 –27, 2024.

et al., 2021). In the latest WebNLG challenge, several pipelines emerged to generate more accurate outputs, incorporating techniques such as data splitting to reduce the input data and backtranslation for low-resource languages (Kumar et al., 2023).

3 GEM 2024 RDF-to-Text Task Description

The GEM Shared Task 2024 focuses on text summarization and RDF-to-Text generation. Our participation is limited to the second task, which involves generating verbalizations from a set of RDF triples. These triples, consisting of a subject, predicate, and object. The shared task provides six files containing RDF triples extracted from the web. Three of these files each contain $1,799$ inputs extracted from WebNLG and are classified as "seen" inputs, because these data could have corresponding gold-standard verbalizations that can be used to train potential statistical or neural systems. The other three files each contain $1,800$ inputs extracted from Wikipedia, for which no gold-standard verbalizations are available online. These inputs are therefore classified as "unseen" inputs. These files contain triples extracted directly from WebNLG and Wikidata, altered triples where the subject or object has been changed and triples with entities generated using LLM prompting. The task is designed for multiple languages, including English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic. In our case, we have chosen to participate in the task using only English.

4 English WebNLG Corpus Description

WebNLG is a corpus containing data units, a set of RDF triples, each paired with one or multiple natural language expressions handwritten by expert annotators, where verbalizations express the semantics of the corresponding data units. For instance:

Data unit:
`(Ajoblanco country Spain)`
`(Ajoblanco ingredient Garlic)`
Verbalization:
Garlic is an ingredient used in Ajoblanco which originates from the country of Spain.

The triples are extracted from 15 different DBpedia categories, including Food, City, and others. The authors selected a wide range of categories to create a resource with a high variety of data (Perez-Beltrachini et al., 2016). The data units contain triples with diverse types of relationships. Among these are chains, where the object of a triple becomes the subject of another triple. There are also siblings, where distinct triples share the same subject. Furthermore, certain data units exhibit mixed relationships, containing both sibling and chain-related triples within them (see Figure 1). The extraction of these triples with varied relationships aimed to capture a wide range of linguistic structures.

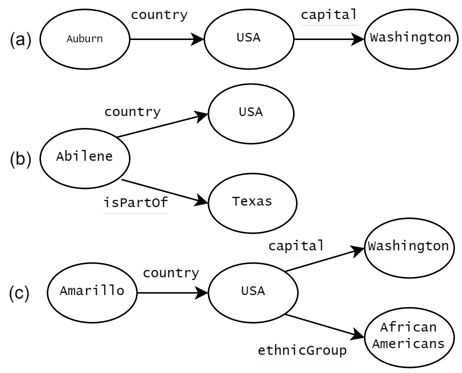


Figure 1: (a) The triples in the data unit are chain-related to each other. `USA` is the subject of the second triple and the object of the first one. (b) The relation between triples in the data unit is defined as sibling. `Abilene` is the subject of all the triples. (c) Some triples in the data unit are sibling-related, while others are chain-related, hence they are referred to as triples in a mixed relation

The latest version of English WebNLG is 3.0, released during the WebNLG challenge in 2020. This version contains $18,812$ data units with $47,195$ verbalizations. The corpus has been divided into training, development, and test sets, each consisting of data units containing 1 to 7 RDF triples.

5 The SGA Pipeline

Our work is based on the SGA pipeline, illustrated in Figure 2, which consists of three main steps: Data Unit Splitting (S), RDF-to-Text generation (G), and Sentence Aggregation (A). While the first step is based on a symbolic deterministic algorithm, the second and third steps rely on LLMs. We chose this modular approach to mitigate the "hallucinations" of LLMs' holistic approach. This was done because we hypothesize that as the amount of input data increases, the performance of LLMs in terms of adequacy and fluency decreases. In the shared task, the provided data units contain up to seven triples. To address these issues, we simplified the problem by dividing the data units into separate sets of triples, which were then verbalized through an RDF-to-Text system and unified using a Sentence

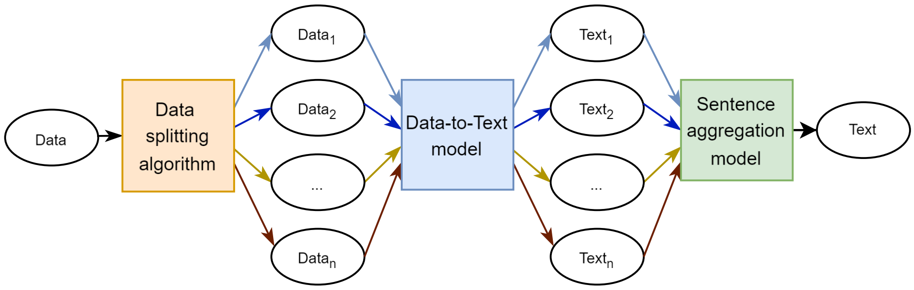


Figure 2: The SGA pipeline we propose begins with structured data, which is divided using a Data Unit Splitting Algorithm. Next, a RDF-to-Text System generates corresponding verbalizations, which are then unified using a Sentence Aggregation System.

Aggregation system.

5.1 Data Unit Splitting Description

Algorithm 1: Data Unit Splitting Algorithm

```
Data: Triples, Max triples per set
Result: Triples sets
subjects_dict = { } objects_dict = { }
foreach t in triples do
    subjects_dict[t.subj].append(t)
    foreach t1 in triples do
        if t.subj == t1.obj then
            objects_dict[t1.obj].append(t1)
        end
    end
end
merged_dict = { }
foreach subj, s_triples in subjects_dict do
    merged_dict[subj] = s_triples
    if subj in objects_dict then
        foreach o_triple in obj_triples[subj]
         do
            if not find(o_triple, merged_dict)
             then
                merged_dict[subj].append(o_triple)
            end
        end
    end
end
return generate_sets(merged_dict,
  max_triples);
```

As described in Section 4, the data units in the test sets provided by the GEM Shared Task could also have different shape types, representing various relationships between them, namely chain, sibling, and mixed type. To reduce the complexity of the data units, i.e., reducing the number of triples in each unit, the main idea is to divide data units into subsets of triples, with a maximum of three triples per set. To achieve this goal, we analyze the shape of each data unit to identify the relationships between triples. Unfortunately, the data units provided by GEM do not contain information about the shape type. Therefore, we created a Splitting algorithm to find the relationship between triples and divide them based on the retrieved information (cf. Algorithm 1). The Splitting algorithm processes triples within a data unit by storing those with identical subjects in `subjects_dict` and those whose objects appear as subjects in other triples in `objects_dict`. It splits each triple into subject, predicate, and object, using the subject as the key in `subjects_dict` and checking if the subject appears as an object in other triples. If so, those triples are added to `objects_dict`. After populating both `subjects_dict` and `objects_dict`, the algorithm merges these dictionaries into a unified structure called `merged_dict`. This involves copying entries from `subjects_dict` into `merged_dict`. For keys that are present in both dictionaries, the algorithm checks if any triple from `subjects_dict` is already listed under that key in `merged_dict`. If a triple is not found in the existing list, it is added to the list, capturing the chain relationships between triples. Once the dictionaries are merged, the algorithm addresses cases where any key in `merged_dict` contains more than three values. It splits these lists into chunks of up to three items each, dividing the triples based on their order. For instance, if there are four triples, the first three are grouped into one subset, while the fourth is placed in a separate subset. This method ensures that no list becomes too large, making the data easier to process and analyze. The choice of three as the chunk size is based on a qualitative analysis of the results from the SGA pipeline. For example, given this data unit with four triples:

```
(Trafford ground Estadio_Hirschi)
(Estadio_Hirschi location Itamarati)
(Trafford league League_One)
(League_One country USA)
```

The dictionaries obtained by the splitting algorithm are:

```
subjects_dict: [
  "Trafford": [
    "Trafford ground Estadio_Hirschi",
    "Trafford league League_One",
  ],
  "Estadio_Hirschi": [
    "Estadio_Hirschi location Itamarati",
  ],
  "League_One": [
    "League_One country USA"
  ]
]
objects_dict: [
  "Estadio_Hirschi": [
    "Trafford ground Estadio_Hirschi",
  ],
  "League_One": [
    "Trafford league League_One"
  ]
]
merged_dict: [
  0: [
    "Trafford ground Estadio_Hirschi",
    "Trafford league League_One",
  ],
  1: [
    "Estadio_Hirschi location Itamarati",
  ],
  2: [
    "League_One country USA"
  ]
]
```

The resulting subsets are:

Set 1:
```
Trafford ground Estadio_Hirschi
Trafford league League_One
```
Set 2:
```
Estadio_Hirschi location Itamarati
```
Set 3:
```
League_One country USA
```

For each set, the corresponding verbalizations will be generated using the approach described in the next section.

5.2 RDF-to-Text Description

Addressing the challenge of converting RDF data into natural language, and following the state-of-the-art, we built an RDF-to-Text system by fine-tuning a LLM, using the English version of WebNLG 3.0 for training. Before fine-tuning, we preprocessed the corpus by removing vertical bars between triple elements, sorting the triples alphabetically by predicate, and then concatenating them. For instance:

Data unit:
```
(Trafford | league | League_One)
(Trafford | nickname | Steve_Bright)
```
Pre-processed data unit:
```
(Trafford league League_One
Trafford nickname Steve_Bright)
```

We chose to fine-tune two different LLMs, specifically LLaMA-2 and Mistral, to evaluate their performance and selected the best-performing model for the GEM Shared Task. We opted for the 7 billion parameter versions of LLaMA-2 and Mistral models.[7][8] Moreover, we used a QLoRA quantization technique (Dettmers et al., 2023) to simplify the fine-tuning process and reduce the computational impact, using the following parameters: the LoRA attention dimension ($lora\_r$) was set to 64, the alpha parameter for LoRA scaling ($lora\_alpha$) was set to 16, and the dropout probability for LoRA layers ($lora\_dropout$) was set to 0.1. Additionally, we fine-tuned the models using only 20% of the dataset. Our training set comprised $7,085$ examples, and the development set included 893 instances. In both models, a single training epoch and a batch size of 4 were used. Furthermore, the following hyperparameters were employed: the maximum gradient norm for gradient clipping was set to 0.3, the initial learning rate for the AdamW optimizer was set to 2×10^{-4}, the weight decay applied to all layers except bias/LayerNorm weights was set to 0.001, and the optimizer used was "paged_adamw_32bit". The learning rate schedule followed a cosine pattern, with the number of training steps set to -1, and a linear warmup ratio of 0.03.

5.3 Sentence Aggregation Description

In this phase, we show how we aggregated the sentences generated in the previous step (RDF-to-Text) to achieve the final verbalization. This was accomplished using an LLM zero-shot prompting technique. After a qualitative assessment of the output with different prompts, the chosen one was:

> "*Instruction="You have to aggregate and paraphrase together the following sentences. You have to generate the result in Italian."*
> *Input: Text1: "...", Text2: "...", ...*
> *Output:""*

We filled this prompt with the texts generated in the previous step, where verbalizations for each

[7] https://huggingface.co/meta-llama/Llama-2-7b
[8] https://huggingface.co/mistralai/Mistral-7B-v0.1

subset of triples were created. For the GEM Shared Task, after a qualitative evaluation of the performance of Mistral-7B and LLaMA-2-7B in the SGA pipeline, we chose to use the former model for both the RDF-to-Text and sentence aggregation steps.

6 Results

In this section, we present the results provided by the organizers for the various systems participating in the GEM Shared Task, evaluated on different subsets of data. The performance of these systems are compared using metrics such as BLEU, METEOR, chrF++, and Bert F1 for the English tasks. The evaluation was conducted using 180 selected data points, each associated with a single reference text. It is important to note that the use of only one reference per data point might lead to lower scores compared to evaluations with multiple references or a larger number of data points.

D2T-1-FA The D2T-1-FA subtask consists of data units directly extracted from the WebNLG test set. In this task, the DipInfo-UniTo system demonstrated excellent performance with a BLEU score of 32.31, making it the top system in this metric. Additionally, it achieved great results across other metrics, ranking among the best systems for this task (see Table 1).

D2T-1-CFA This subtask involves switching entities in the data units extracted from WebNLG (e.g., replacing a person entity with another person entity, a date with another date, etc.). DipInfo-UniTo achieved the highest scores across all metrics, surpassing the other participants by a significant margin, making it the best system for this task (see Table 2).

D2T-1-FI In the D2T-1-FI subtask, which is the most challenging of all the D2T-1 dataset, data units were first extracted from WebNLG and then modified with entities generated by an LLM. The DipInfo-UniTo system achieved the highest scores across all metrics, maintaining a significant lead over the other systems, like in the previous subtask (see Table 3).

D2T-2-FA This subtask involves data units directly extracted from Wikidata. The DipInfo-UniTo system achieved the highest score on the Bert F1 metric (0.937) and ranked as the second-best system in the other metrics, just behind SaarLST (see Table 4)

D2T-2-CFA The D2T-2-CFA subtask features data units extracted from Wikidata with swapped entities. The DipInfo-UniTo system achieved a BLEU score of 32.01, the highest among all systems. It was also the second-best in the other metrics, showcasing its strong performance. Specifically, SaarLST outperformed DipInfo-UniTo in METEOR and chrF++, while DCU-NLG-PBN excelled in the Bert F1 metric (see Table 5).

D2T-2-FI Finally, the D2T-2-FI subtask involves data units from Wikidata with entities generated by an LLM. The DipInfo-UniTo system demonstrated strong performance, achieving a BLEU score of 21.26, the highest among all participants. It also ranked second in the other metrics for this task, being outperformed by SaarLST in METEOR and chrF++, and by DCU-NLG-PBN in the Bert F1 metric (see Table 6)

In conclusion, the DipInfo-UniTo system has proven to be highly competitive across all tasks, frequently achieving the highest scores among participants and only falling slightly short in other cases, demonstrating excellent generalization ability across various datasets.

7 Conclusion

The main objective of this work was to enhance the performance of LLMs in RDF-to-Text generation. To achieve this, we employed NLG techniques with LLMs to develop the SGA pipeline designed to simplify the task and improve the quality of the outputs. To demonstrate the effectiveness of this technique, we compared the performance of LLaMA-2 and Mistral models both with fine-tuning and within the SGA pipeline. The results show that our approach improves performance on the RDF-to-Text task. The developed system demonstrated strong competitiveness across all tasks in the GEM 2024, achieving the highest scores in some cases while narrowly missing out in others. This performance underscores its good ability to generalize across various datasets. Future work could involve refining this technique by fine-tuning the models to better specialize in sentence aggregation, developing a more sophisticated data splitting algorithm, and integrating additional NLG techniques to produce more fluent and accurate text.

8 Limitations

The main limitation of our work was the limited computational resources available. To achieve better results, it would be necessary to use the entire WebNLG 3.0 corpus for fine-tuning the models, employ larger LLMs, and analyze performance by adjusting training hyperparameters to identify the configurations that yield the best performance.

References

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, and William Soto Martinez. 2023. The 2023 webnlg shared task on low resource languages. overview and evaluation results (webnlg 2023). In *Association for Computational Linguistics*, page 55–66.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the webnlg corpus. In *Association for Computational Linguistics*, page 171–176.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. jiang 7b. *Preprint*, arXiv:2310.06825.

Nalin Kumar, Saad Obaid Ul Islam, and Ondrej Dusek. 2023. Better translation + split and generate for multilingual RDF-to-text (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 73–79, Prague, Czech Republic. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building rdf content for data-to-text generation. In *The COLING 2016 Organizing Committee*, pages 1493–1502.

Anastasia Shimorina, Elena Khasanova, and Claire Gardent. 2019. Creating a corpus for russian data-to-text generation using neural machine translation and post-editing. In *Association for Computational Linguistics*, pages 44–49.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Qingyun Wang, Semih Yavuz, Xi Victoria Lin, Heng Ji, and Nazneen Rajani. 2021. Stage-wise fine-tuning for graph-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 16–22, Online. Association for Computational Linguistics.

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-ADAPT-modPB	30.78	0.332	0.555	**0.935**
DCU-NLG-PBN	29.08	0.33	0.555	0.933
DCU-NLG-Small	27.0	0.314	0.537	0.93
DipInfo-UniTo	**32.31**	0.346	0.58	0.933
OSU-CompLing	30.03	0.335	0.566	0.932
RDFpyrealb	26.37	0.331	0.551	0.928
SaarLST	29.7	**0.347**	**0.581**	0.931

Table 1: Metrics scores on the D2T-1-FA English task (1 reference text per data point).

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-ADAPT-modPB	26.98	0.299	0.515	0.924
DCU-NLG-PBN	25.2	0.297	0.513	0.923
DCU-NLG-Small	22.98	0.279	0.488	0.918
DipInfo-UniTo	**29.01**	**0.315**	**0.543**	**0.926**
OSU-CompLing	24.45	0.293	0.514	0.92
RDFpyrealb	21.67	0.291	0.495	0.918
SaarLST	23.48	0.307	0.524	0.921

Table 2: Metrics scores on the D2T-1-CFA English task (1 reference text per data point).

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-ADAPT-modPB	26.54	0.318	0.539	0.921
DCU-NLG-PBN	26.02	0.322	0.549	0.92
DCU-NLG-Small	20.85	0.292	0.507	0.914
DipInfo-UniTo	**28.24**	**0.342**	**0.587**	**0.924**
OSU-CompLing	21.44	0.306	0.537	0.915
RDFpyrealb	21.97	0.31	0.527	0.917
SaarLST	20.76	0.331	0.557	0.917

Table 3: Metrics scores on the D2T-1-FI English task (1 reference text per data point).

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-NLG-PBN	23.96	0.295	0.49	0.936
DCU-NLG-Small	19.48	0.26	0.438	0.925
DipInfo-UniTo	27.22	0.304	0.512	**0.937**
OSU-CompLing	24.97	0.295	0.496	0.934
RDFpyrealb	19.97	0.287	0.479	0.921
SaarLST	**28.25**	**0.32**	**0.538**	0.934

Table 4: Metrics scores on the D2T-2-FA English task (1 reference text per data point).

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-NLG-PBN	30.34	0.348	0.581	**0.937**
DCU-NLG-Small	24.9	0.3	0.51	0.923
DipInfo-UniTo	**32.01**	0.354	0.592	0.936
OSU-CompLing	27.06	0.334	0.567	0.93
RDFpyrealb	25.05	0.335	0.561	0.923
SaarLST	26.47	**0.359**	**0.597**	0.929

Table 5: Metrics scores on the D2T-2-CFA English task (1 reference text per data point).

System ID	BLEU	METEOR	chrF++	Bert F1
DCU-NLG-PBN	20.46	0.3	0.49	**0.924**
DCU-NLG-Small	16.88	0.267	0.442	0.914
DipInfo-UniTo	**21.26**	0.307	0.502	0.923
OSU-CompLing	16.9	0.282	0.475	0.917
RDFpyrealb	16.28	0.286	0.472	0.916
SaarLST	20.16	**0.315**	**0.518**	0.919

Table 6: Metrics scores on the D2T-2-FI English task (1 reference text per data point).

DCU-ADAPT-modPB at the GEM'24 Data-to-Text Generation Task: Model Hybridisation for Pipeline Data-to-Text Natural Language Generation

Chinonso Cynthia Osuji[♡†], **Rudali Huidrom**[♡†], **Kolawole John Adebayo**[♡], **Thiago Castro Ferreira**[◇], **Brian Davis**[♡]
ADAPT Research Centre, Dublin City University, Ireland[♡]
aiXplain, USA[◇]
chinonso.osuji@adaptcentre.ie rudali.huidrom@adaptcentre.ie
kolawole.adebayo@adaptcentre.ie thiago@aixplain.com
brian.davis@adaptcentre.ie

Abstract

In this paper, we present our approach to the GEM Shared Task at the INLG'24 Generation Challenges, which focuses on generating data-to-text in multiple languages, including low-resource languages, from WebNLG triples. We employ a combination of end-to-end and pipeline neural architectures for English text generation. To extend our methodology to Hindi, Korean, Arabic, and Swahili, we leverage a neural machine translation model. Our results demonstrate that our approach achieves competitive performance in the given task.

1 Introduction

The GEM 2024 Shared Task (Mille et al., 2024) aims to advance summarisation and data-to-text (D2T) generation, with a particular focus on enhancing multilingual capabilities. The D2T task (Reiter and Dale, 1997) involves generating coherent natural language text from structured data in the form of Wikidata and WebNLG datasets, which are organised as triples consisting of a subject, predicate, and object. The goal of the tasks is to comprehensively evaluate and improve the ability of systems to interpret and generate text from RDF triples, assess their general knowledge, and produce texts in factual (FA), counterfactual (CFA), and fictional (FI) scenarios.

The dominance of English in D2T generation presents a considerable challenge, highlighting the need for research to support effective multilingual generation, particularly for languages with diverse morphological structures and distinct word order characteristics. The GEM 2024 Shared Task addresses this challenge by including English alongside other languages such as Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic, which are low-resource in the D2T setting. This task aims to enhance the adaptability and robustness of different systems across varied linguistic frameworks for text generation from structured data.

In this submission, we focus on the D2T generation aspect of the task using the WebNLG dataset (Castro Ferreira et al., 2020). Our approach combines end-to-end and pipeline neural architectures to generate English text, while also fine-tuning a state-of-the-art open-source Flan-T5 and Mistral-7B large language models (LLMs) for generating text in low-resource languages. Our approach aims to further the understanding of how various architectures can be optimised for multilingual D2T generation. Our methodology demonstrates competitive performance and contributes substantial insights and advancements to the field of multilingual D2T generation. The code and results are available[1].

2 Related Work

The field of data-to-text generation has undergone significant transformations, evolving from traditional pre-neural approaches that relied on handcrafted rules, templates, and statistical models (Reiter and Dale, 1997; Erdem et al., 2022) to modern deep learning architectures. These advanced models are trained to identify and replicate the relationships between structured data and its corresponding textual outputs. The introduction of end-to-end systems, particularly pre-trained language models (PLMs), has substantially improved the processing of textual sequences in data-to-text tasks (Kale and Rastogi, 2020; Ribeiro et al., 2021). However, despite their advanced capabilities, these systems often struggle with content selection and maintaining fidelity due to the opaqueness and complexity inherent to deep learning models and the data-to-text generation task(Moryossef et al., 2019).

[†] The first two authors made equal contributions to all aspects of the work, the order in which they appear was decided arbitrarily.

[1] https://github.com/NonsoCynthia/GEM2024_ST

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 66–75
September 23 –27, 2024.

A recent example of methodological advancement in this field is showcased in the 2023 WebNLG Shared Task on Low Resource Languages, where many participants employed NLG+MT (Natural Language Generation plus Machine Translation) pipeline approach (Cripwell et al., 2023). For instance, some participants implemented systems which generate English text from RDF graphs using a PLM fine-tuned on the WebNLG 2020 dataset, followed by translation into various languages using a machine translation (MT) model (Aditya Hari et al., 2023; Kumar et al., 2023). This approach showcases the potential of combining NLG and MT models for effective multilingual data-to-text generation.

Similarly, Lorandi and Belz (2023) proposed a novel approach that utilises large language models (GPT-3.5 and GPT-4) for prompt-based generation without additional training. They tested two methods: direct generation in under-resourced languages and generation in English followed by translation using Google Translate. In our research, we build upon these methodologies and incorporate a 3-stage pipeline neural architecture, as in Figure 1, inspired by Ferreira et al. (2019). However, we modify the approach by integrating only the first two stages of ordering and structuring, followed by the final stage of surface realisation. This approach aims to optimise the use of large language models for multilingual data-to-text generation.

3 Methodology

In this section, we outline the methodologies employed to address the generation challenge for the languages English (en), Hindi (hi), Korean (ko), Arabic (ar), and Swahili (sw). Our experimental setup is as follows:

3.1 Data

We utilised the enhanced WebNLG dataset (Castro Ferreira et al., 2018) for fine-tuning the ordering and structuring stages in the intermediate phases of the pipeline neural architecture. For fine-tuning the Mistral7b model, we used the WebNLG'17 dataset (Gardent et al., 2017). Finally, we evaluate the performance of the fine-tuned models using the GEM 2024 Shared Task D2T dataset, which encompasses factual, fictional, and counterfactual domains, each containing 1779 RDF triple sets.

3.2 System Description

The GEM 2024 Shared Task focuses on summarisation and data-to-text (D2T) generation, with a particular emphasis on multilingual capabilities. For this task, only testing data is provided, consisting of three parallel datasets: Factual (FA), Counterfactual (CFA), and Fictional (FI). The FA dataset uses original triples from WebNLG'20 data (Castro Ferreira et al., 2020) and Wikidata (Vrandečić and Krötzsch, 2014), while the CFA dataset replaces entities in the factual dataset with similar-class entities, e.g., by swapping person names, dates, etc. The FI dataset substitutes entities in the factual dataset with fabricated entities generated by large language models (LLMs). Our work concentrates exclusively on data-to-text generation of triples from WebNLG.

Pipeline Neural Architecture: We designed a pipeline neural architecture, depicted in Figure 1, which leverages the fine-tuned Flan-T5-*large* model (Chung et al., 2022) to perform ordering and structuring tasks on the enhanced WebNLG 2017 dataset (Castro Ferreira et al., 2018). The Flan-T5 model is initially fine-tuned separately for ordering and structuring tasks using a subset of the enhanced WebNLG dataset. As shown in Figure 2, the pipeline architecture takes test set triples (FA, CFA, FI) as input and passes them through the ordering model to determine their verbalisation sequence. The ordered triples are then mapped to their corresponding entities (subjects and objects values) and fed into the structuring model. The structuring model organises the entities into coherent sentences, marking sentence boundaries with [SNT] and [/SNT] tags, while ensuring accurate entity mappings. Predicates serve as pointers during this process, linking to their respective triples after generation.

Finally, for surface realisation, we integrated prompt-based models, including Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) and GPT-4 Turbo (Ye et al., 2023; Achiam et al., 2023). The structured outputs are fed into these prompt-based models to generate the final text. The overall workflow is presented in Figure 2.

Parameter Efficient Instruction Fine-Tuning: Our second setup employs parameter efficient fine-tuning (PEFT) (Houlsby et al., 2019) for instruction tuning of the selected models. Specifically, we utilise LORA (Hu et al., 2021), which inte-

	BLEU ↑	METEOR ↑	ChrF++ ↑	TER ↓	BERT_P ↑	BERT_R ↑	BERT_F1 ↑
StructGPT4	**49.80**	0.40	0.655	**0.450**	**0.958**	0.953	**0.955**
GPT4	42.823	**0.418**	**0.677**	0.548	0.948	**0.957**	0.952
Mistral	37.552	0.378	0.623	0.559	0.943	0.949	0.945
StructMistral	35.493	0.353	0.584	0.578	0.940	0.941	0.940
FinetunedMistral	31.070	0.29	0.513	0.630	0.913	0.916	0.914

Table 1: Automatic metrics results of our systems for factual (FA) English test set. Bold and underlined results denote the best and the second best ones respectively.

	FACTUAL				
	Arabic	**Hindi**	**Korean**	**Swahili**	**English**
StructGPT4	0.499	0.425	0.581	0.612	0.629
GPT4	0.546	0.478	**0.633**	**0.627**	**0.636**
Mistral	**0.558**	0.445	0.608	0.613	0.625
StructMistral	0.498	**0.615**	0.581	0.612	0.615
FinetunedMistral	0.498	0.276	0.433	0.574	0.551
	COUNTERFACTUAL				
	Arabic	**Hindi**	**Korean**	**Swahili**	**English**
StructGPT4	0.511	0.406	0.576	0.567	0.49
GPT4	**0.551**	**0.448**	**0.613**	0.571	**0.518**
Mistral	0.519	0.415	0.584	0.580	0.471
StructMistral	0.479	0.374	0.542	**0.581**	0.441
FinetunedMistral	0.308	0.239	0.372	0.556	0.254
	FICTIONAL				
	Arabic	**Hindi**	**Korean**	**Swahili**	**English**
StructGPT4	0.508	0.408	0.589	0.554	**0.499**
GPT4	0.137	0.062	0.180	**0.564**	0.108
Mistral	**0.530**	**0.428**	**0.602**	0.559	0.484
StructMistral	0.494	0.397	0.575	0.563	0.460
FinetunedMistral	0.300	0.231	0.369	0.532	0.238

Table 2: COMET metrics results of our systems for FA, CFA and FI test set for all the languages. Bold and underlined results denote the best and the second best ones respectively.

grates trainable adapters in the form of low-rank decomposition matrices into chosen layers of a transformer model. To enhance the diversity of our training data, we designed a template that produces 10 rewritten instructions for each original instruction. These re-written instructions are worded differently, but convey the same meaning or action trigger, allowing the fine-tuned model to align more robustly to varied instructions and improve its ability to generalise to new, unseen inputs. We use the the WebNLG'17 corpus (Gardent et al., 2017) for the model fine-tuning. We then combine the fine-tuned model with the base model, leveraging both the specialised fine-tuning and the broad knowledge inherent from pretraining. This composite model is tested with 5 examples from the WebNLG corpus, along with our newly created dataset.

In-Context Learning: In our final setup, we utilised the in-context learning (Zhao et al., 2023; Yang et al., 2024) capabilities of the selected models, namely Mistral7b, and GPT-4, for text generation tasks. We performed few-shot prompting using

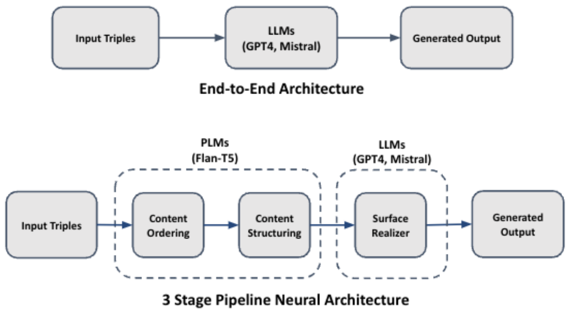


Figure 1: System Description.

five triples randomly selected from the WebNLG corpus. The prompt designs used in our experiments are presented in Appendix A.

3.3 Machine Translation Model

The English outputs generated by the systems described in Section 3.2 were translated into Hindi, Korean, Arabic, and Swahili using specialised machine translation models. For the translation of Korean, Arabic, and Swahili, we utilised the open-source Command-R-Plus model developed by Cohere (Üstün et al., 2024). Specifically, we utilised the 4-bit quantised version which is available on the HuggingFace model hub[2]. The translation into Hindi was performed using the IndicTrans2 model (Gala et al., 2023), which is also an open-source transformer-based multilingual NMT model specifically trained for all 22 officially recognised Indic languages. Our selection of the two multilingual models was based on their open-source availability and their relative performance in the languages covered in our experiments. We conducted preliminary limited testing to evaluate their performance by having native language speakers assess the quality of the translated text. Their feedback informed our decision to use these translation models for our experiments.

[2] https://huggingface.co/CohereForAI/c4ai-command-r-plus-4bit

4 Results

In our results' naming convention, "Struct" denotes the pipeline architecture system that utilises structured triples for generation. "FinetunedMistral" refers to the fine-tuned Mistral-7B-Instruct system, while systems without these acronyms represent direct generation using the base models within the end-to-end architecture.

The results from the evaluation in Table 1 provide valuable insights into the strengths and weaknesses of the different models across various automatic metrics within the English language in the FA dataset. StructGPT4 achieved the highest scores in BLEU (49.80), TER (0.45), BERT_P (0.958), and BERT_F1 (0.955) for English. Following this, GPT4 consistently emerges as the most versatile and high-performing model, excelling in a wide range of languages (Arabic, Hindi, Korean, Swahili, and English) and domains (FA, CFA, FI). For instance, in the FA English test set, GPT4 achieves top scores in METEOR (0.418), ChrF++ (0.677), and BERT_F1 (0.952), underscoring its ability to produce translations that are both semantically accurate and closely aligned with reference texts.

Furthermore, we employed the COMET metric (Rei et al., 2020), a neural evaluation model specifically designed to predict quality scores for translations. COMET is known for demonstrating a strong correlation with human judgement and is capable of performing reference-less evaluations. This capability makes COMET particularly well-suited for assessing our results in non-English languages within the FA dataset, as well as for all languages in the CFA and FI datasets, where reference translations are not yet available. The results of our evaluation using COMET are presented in Table 2. The results indicate that GPT-4 consistently performs well, particularly in the FA and CFA datasets, achieving the highest scores in English (0.636 for FA, 0.518 for CFA) and in several other languages (see Table 2). However, GPT-4 struggles in the FI dataset, especially in Arabic, Hindi, and Korean, with scores as low as 0.137 in Arabic. Mistral shows strong performance across all datasets, particularly excelling in the FI dataset, where it achieves the highest scores in Arabic (0.530), Hindi (0.428), and Korean (0.602). StructGPT4 also performs well, leading in the FI dataset with a score of 0.499 in English, and shows strong results in other datasets, especially in Arabic and Korean. StructMistral is competitive in Swahili, particularly in the CFA dataset (0.583), but generally ranks second in most other cases. In contrast, FinetunedMistral underperforms across all languages and datasets, with notably low scores, such as 0.254 in English for the CFA dataset. Overall, GPT-4 and Mistral emerge as the top-performing models for the COMET metrics, but their effectiveness varies depending on the dataset and language, highlighting the importance of context in model performance.

5 Analysis and Discussion

In this analysis, we highlight the factors which may have contributed to the varying performances of the models in our experiments.

First, the underlying architecture and training data play a critical role. We observe that our GPT4-based systems benefits from extensive training on a large and diverse dataset, which likely contributes to its consistent performance across different languages and domains. The robustness of its architecture allows it to handle a wide range of tasks effectively. However, we observed a decline in performance within the FI dataset. Upon manual inspection, we found that the system generated text with the correct entities but often rejected certain entity claims in the dataset, leading to its overall poor performance in this category.

Second, the fine-tuning process and the nature of the tasks significantly influence performance. StructGPT4, for instance, is fine-tuned with a focus on specific tasks (i.e., ordering and structuring) requiring precision and the handling of complex or nuanced content, which explains its superior performance in BLEU and TER, especially in FA English text generation.

Third, language-specific optimisations or model adaptations can lead to better performance in certain languages. Mistral shows strong results in Korean and Swahili, which may indicate that it has been trained or optimised for these specific languages, allowing it to outperform GPT4 and StructGPT4 in these contexts.

Fourth, the evaluation metrics themselves might favour certain models depending on how they align with the strengths of each model. For example, StructGPT4 performs better in BLEU and TER, metrics that emphasise precision and reduced errors, while GPT4 excels in METEOR and ChrF++, which also account for semantic accuracy and fluency.

These factors highlight the importance of selecting models based on the specific requirements of the task, considering not only the general capabilities of the model but also how well it has been optimised or fine-tuned for particular languages and tasks. To fully harness the aggregate benefits of the various factors influencing the performance of models as identified in our experiment, future work should focus on conducting a comprehensive exploration of each aspect. This may involve:

- **Experimental Design Optimisation:** Investigating different architectural designs, such as combining structured and prompt-based approaches, to identify the most effective methods for enhancing model performance.
- **Fine-tuning Strategies:** Exploring fine-tuning techniques that can better balance the retention of learned general capabilities and adaptation to specific tasks, thereby minimising the risk of overfitting and improving model generalisation.
- **Dataset Selection:** Examining the impact of training data on model performance by comparing the performance of these models when finetuned with canonical datasets from multiple GEM and WebNLG competitions, thereby gaining insights on dataset diversity and size on model adaptation and generalisation for D2T generation tasks.
- **Evaluation Methods:** Enhancing evaluation methodologies by integrating both automatic and human evaluations, ensuring a more accurate and nuanced assessment of model performance. This may involve developing new metrics that can better capture the subtleties of generated text in the context of D2T tasks.

6 Conclusion and Future Directions

In conclusion, this paper presents the methodologies and automatic evaluation results of our submission to the GEM 2024 tasks. The evaluation results highlight the strengths of different models across various metrics and languages. StructGPT4 stands out in producing precise translations with fewer errors, especially in English, outperforming GPT4 in metrics like BLEU and TER. GPT4, however, proves to be the most versatile and high-performing model across multiple languages and domains, excelling in METEOR, ChrF++, BERT_F1, and COMET metrics, although it shows limitations in generating text within the FI task.

Mistral demonstrates strong performance in languages such as Korean, Hindi, and Arabic, particularly within the FI task, while StructMistral excels in Swahili CFA tasks. These findings suggest that while GPT4 is the most reliable general-purpose model, StructGPT4, due to its incorporation of task splitting and pipelining, is better suited for tasks requiring minimal errors, high accuracy, and attention to detail. Meanwhile, Mistral and StructMistral offer valuable performance in specific applications, indicating their potential for specialised use cases.

In order to gain a more comprehensive understanding of our systems' performance, we look forward to the availability of human evaluation results, which will provide valuable insights and enable us to draw further conclusions. Moreover, we plan to further explore the impact of advanced fine-tuning methods with preference-based learning, such as recent state-of-the-art frameworks like DPO (Rafailov et al., 2024), KTO (Ethayarajh et al., 2024), SPPO (Wu et al., 2024) and the REINFORCE (Ahmadian et al., 2024) preference optimisation. These methods have shown promise in improving model alignment and generation performance, and we believe they could be valuable additions to our existing systems.

We will also investigate the possible impact of data selection and prompt engineering methods on optimising our existing systems. Studies, for example in (Shen, 2024; Liu et al., 2024) have shown that carefully selecting and preparing high-quality data for LLM finetuning often leads to improvement in model performance. This is because high-quality data allows the model to learn from relevant and accurate examples, which is crucial for fine-tuning the model's parameters and achieving optimal performance.

Lastly, we are keen on investigating the development of an end-to-end framework that encompasses ordering, structuring, and text generation collectively. This would allow us to streamline our pipeline and potentially improve the overall performance of our systems.

Ethics Statement

We adhered to the structure of the ARR responsible research checklist. The risk associated with this study was minimal.

Acknowledgments

Osuji's work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223; and ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology at Dublin City University under Grant No 13/RC/2106_P2. Huidrom's work on this project was supported by the Faculty of Engineering and Computing, DCU, via a PhD grant. Adebayo's contribution on this work was supported by Enterprise Ireland's CareerFit-Plus Co-fund and the European Union's Horizon 2020 research and innovation programme Marie Skłodowska-Curie Grant No. 847402. We would like to thank Prof. Anya Belz for her kind guidance and discussions throughout the course of this paper.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kancharla Aditya Hari, Bhavyajeet Singh, Anubhav Sharma, and Vasudeva Varma. 2023. WebNLG challenge 2023: Domain adaptive machine translation for low-resource multilingual RDF-to-text generation (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 93–94, Prague, Czech Republic. Association for Computational Linguistics.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

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.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 171–176, Tilburg University, The Netherlands. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint*.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, and William Soto Martinez. 2023. The 2023 WebNLG shared task on low resource languages. overview and evaluation results (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 55–66, Prague, Czech Republic. Association for Computational Linguistics.

Erkut Erdem, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii, Oleksii Turuta, Aykut Erdem, Iacer Calixto, et al. 2022. Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning. *Journal of Artificial Intelligence Research*, 73:1131–1207.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 552–562.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea

Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Nalin Kumar, Saad Obaid Ul Islam, and Ondrej Dusek. 2023. Better translation + split and generate for multilingual RDF-to-text (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 73–79, Prague, Czech Republic. Association for Computational Linguistics.

Ziche Liu, Rui Ke, Feng Jiang, and Haizhou Li. 2024. Take the essence and discard the dross: A rethinking on data selection for fine-tuning large language models. *arXiv preprint arXiv:2406.14115*.

Michela Lorandi and Anya Belz. 2023. Data-to-text generation for severely under-resourced languages with GPT-3.5: A bit of help needed from Google Translate (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 80–86, Prague, Czech Republic. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. 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 2267–2277.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

Ming Shen. 2024. Rethinking data selection for supervised fine-tuning. *arXiv preprint arXiv:2402.06094*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *Preprint*, arXiv:2402.07827.

A Prompt Design

Table 4 illustrates our prompt design for English with five examples each for with and without structured data. We report two types of prompts, one

Input Triples:
[TRIPLE] Bananaman broadcastedBy BBC [/TRIPLE] [TRIPLE] Bananaman creator John_Geering [/TRIPLE] [TRIPLE] Bananaman firstAired "1983-10-03" [/TRIPLE] [TRIPLE] Bananaman lastAired "1986-04-15" [/TRIPLE] [TRIPLE] Bananaman starring Tim_Brooke-Taylor [/TRIPLE]

Ordering Output:
broadcastedBy firstAired lastAired creator starring

Input Triples after mapping:
[TRIPLE] Bananaman broadcastedBy BBC [/TRIPLE] [TRIPLE] Bananaman firstAired "1983-10-03" [/TRIPLE] [TRIPLE] Bananaman lastAired "1986-04-15" [/TRIPLE] [TRIPLE] Bananaman creator John_Geering [/TRIPLE] [TRIPLE] Bananaman starring Tim_Brooke-Taylor [/TRIPLE]

Structuring Output:
[SNT] broadcastedBy firstAired lastAired [/SNT] [SNT] creator starring [/SNT]

Input Triples after mapping and removing the [TRIPLE] start and [TRIPLE] end tags:
[SNT] Bananaman broadcastedBy BBC, Bananaman firstAired "1983-10-03", Bananaman lastAired "1986-04-15" [/SNT] [SNT] Bananaman creator John_Geering, Bananaman starring Tim_Brooke-Taylor [/SNT]

Surface Realizer Output:
Bananaman was shown on the BBC, first airing on October 3, 1983 and the final broadcast being April 15, 1986. It was created by John Geering and starred Tim Brooke Taylor.

Figure 2: Pipeline Neural Architecture Outputs

Factual Dataset Result							
	Models	**0**	**1**	**2**	**3**	**4**	**Average**
English	**StructGPT4**	0.8132	0.8096	0.7654	0.3805	0.3781	0.629
	GPT4	0.8189	0.815	0.7713	0.3874	0.3851	0.636
	Mistral	0.8035	0.8005	0.7583	0.383	0.3808	0.625
	StructMistral	0.7855	0.7838	0.7425	0.3832	0.3809	0.615
	FinetunedMistral	0.6909	0.6884	0.6525	0.3619	0.3596	0.551
Arabic	**StructGPT4**	0.6228	0.6208	0.5919	0.3317	0.3296	0.499
	GPT4	0.684	0.6821	0.6509	0.357	0.3552	0.546
	Mistral	0.6817	0.6807	0.65	0.3902	0.3884	0.558
	StructMistral	0.6046	0.6043	0.5755	0.3521	0.3496	0.497
	FinetunedMistral	0.605	0.6048	0.5758	0.3521	0.3497	0.498
Hindi	**StructGPT4**	0.5061	0.5083	0.4859	0.3122	0.3102	0.425
	GPT4	0.5847	0.5854	0.5588	0.3307	0.3291	0.478
	Mistral	0.5395	0.542	0.5177	0.3145	0.313	0.445
	StructMistral	0.4818	1.4841	0.4649	0.3232	0.3211	0.615
	FinetunedMistral	0.3196	0.3209	0.2101	0.2665	0.2646	0.276
Korean	**StructGPT4**	0.6828	0.6817	0.6549	0.4426	0.4409	0.581
	GPT4	0.7473	0.7466	0.7196	0.4777	0.4759	0.633
	Mistral	0.7205	0.7196	0.6925	0.4555	0.4541	0.608
	StructMistral	0.6704	0.6705	0.6466	0.4602	0.4581	0.581
	FinetunedMistral	0.4701	0.4696	0.4572	0.385	0.3832	0.433
Swahili	**StructGPT4**	0.6513	0.6504	0.6389	0.5602	0.5593	0.612
	GPT4	0.6671	0.6663	0.6544	0.5742	0.5733	0.627
	Mistral	0.652	0.6514	0.6402	0.5621	0.5614	0.613
	StructMistral	0.6485	0.6482	0.6379	0.5639	0.5629	0.612
	FinetunedMistral	0.6033	0.6026	0.5935	0.5365	0.5356	0.574

Table 3: Factual dataset COMET results of the individual reference texts (0, 1, 2, 3, & 4) for evaluation.

for GPT4 model and the other for the Mistral-7B-Instruct model.

Table 5 presents our prompt design for translating English to Arabic, Korean and Swahili using command-r-plus-4bit model from Cohere AI. We provide five examples each for the respective languages.

System instruction	"You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being socially unbiased and safe. If you're unsure about an answer, it's okay to skip it, and please ensure not to provide incorrect information. Additionally, responses should be concise and informative."
User instruction	"I would like you to generate a fluent and concise summaries or text in English based on the triples provided. Below you may find examples of the input triples and the expected summary outputs. Do not omit any triple information in the text or include any information that cannot be directly inferred from the given triples."
Data examples	**1:** *'Input'*: 'Uruguay leader Tabaré_Vázquez, Uruguay leader Raúl_Fernando_Sendic_Rodríguez, Alfredo_Zitarrosa deathPlace Montevideo, Montevideo country Uruguay', *'Output'*: 'Alfredo Zitarrosa died in Montevideo, Uruguay which is led by Raúl Fernando Sendic Rodríguez and Tabaré Vázquez.', **2:** *'Input'*: 'Angola_International_Airport location Ícolo_e_Bengo, Ícolo_e_Bengo country Angola, Angola_International_Airport cityServed Luanda, Ícolo_e_Bengo isPartOf Luanda_Province, Angola_International_Airport elevationAboveTheSeaLevelInMetres 159', *'Output'*: 'Angola International Airport is located at Ícolo e Bengo in Luanda province, Angola. The Airport is situated 159 meters above sea level and serves the city of Luanda.', **3:** *'Input'*: 'United_Petrotrin_F.C. ground Palo_Seco, Akeem_Adams club Trinidad_and_Tobago_national_under-20_football_team, Akeem_Adams club United_Petrotrin_F.C.', *'Output'*: 'Akeem Adams, who plays for the Trinidad and Tobago national under-20 football team previously played for United Petrotrin FC whose ground is at Palo Seco.', **4:** *'Input'*: 'William_Anders selectedByNasa 1963, William_Anders nationality United_States, William_Anders birthDate "1933-10-17", William_Anders occupation Fighter_pilot, William_Anders birthPlace British_Hong_Kong, William_Anders mission Apollo_8', *'Output'*: 'The United States fighter pilot William Anders was born in British Hong Kong on the 17th of October, 1933. In 1963, he was chosen by NASA and became a crew member on Apollo 8.', **5:** *'Input'*: "Dead_Man's_Plack location England, England ethnicGroup British_Arabs, England capital London, Dead_Man's_Plack dedicatedTo Æthelwald,_Ealdorman_of_East_Anglia, England language Cornish_language, England religion Church_of_England, Dead_Man's_Plack material Rock_(geology)", *'Output'*: "The capital of England is London where we can find the Dead Man's Plack which is made of stone. The Plack is dedicated to Æthelwald, Ealdorman of East Anglia. Cornish language is spoken in England and it has an established religion called the Church of England. One of the ethnic groups found in that country is the British Arabs."
Source	Input triple(s) from the test set. E.g. Andra_(singer) genre Rhythm_and_blues, Andra_(singer) background "solo_singer", Rhythm_and_blues derivative Disco
(Structured) Data examples	**1:** *'Input'*: "[SNT] [TRIPLE] Atatürk_Monument_(İzmir) material 'Bronze' [/TRIPLE] [TRIPLE] Atatürk_Monument_(İzmir) inaugurationDate '1932-07-27' [/TRIPLE] [/SNT] [SNT] [TRIPLE] Atatürk_Monument_(İzmir) location Turkey [/TRIPLE] [TRIPLE] Turkey capital Ankara [/TRIPLE] [TRIPLE] Turkey largestCity Istanbul [/TRIPLE] [/SNT] [SNT] [TRIPLE] Turkey leaderName Ahmet_Davutoğlu [/TRIPLE] [TRIPLE] Turkey currency Turkish_lira [/TRIPLE] [/SNT]", *'Output'*: "The Atatürk Monument is a bronze monument inaugurated on 27th July, 1932, in Izmir. It is found in Turkey, a country which has Ankara as its capital and Istanbul as its largest city. The leader of Turkey is called Ahmet Davutoğlu, and the currency is the Turkish lira.", **2:** *'Input'*: "[SNT] [TRIPLE] Turkey capital Ankara [/TRIPLE] [TRIPLE] Turkey largestCity Istanbul [/TRIPLE] [/SNT] [SNT] [TRIPLE] Turkey leader Ahmet_Davutoğlu [/TRIPLE] [TRIPLE] Turkey currency Turkish_lira [/TRIPLE] [/SNT] [SNT] [TRIPLE] Atatürk_Monument_(İzmir) location Turkey [/TRIPLE] [/SNT]", *'Output'*: "The capital of Turkey is Ankara, although the largest city is Istanbul. The leader of Turkey is Ahmet Davutoglu and the currency is known as the Turkish lira. The Ataturk monument is located within the country.", **3:** *'Input'*: "[SNT] [TRIPLE] Antwerp_International_Airport cityServed Antwerp [/TRIPLE] [TRIPLE] Antwerp country Belgium [/TRIPLE] [TRIPLE] Belgium leaderName Philippe_of_Belgium [/TRIPLE] [TRIPLE] Belgium language French_language [/TRIPLE] [/SNT]", *'Output'*: "Antwerp is served by Antwerp International Airport and is a popular tourism destination in Belgium where the leader is Philippe of Belgium and the French language is spoken.", **4:** *'Input'*: "[SNT] [TRIPLE] AWH_Engineering_College state Kerala [/TRIPLE] [TRIPLE] AWH_Engineering_College country India [/TRIPLE] [TRIPLE] AWH_Engineering_College established 2001 [/TRIPLE] [/SNT] [SNT] [TRIPLE] India river Ganges, India largestCity Mumbai [/TRIPLE] [/SNT] [SNT] Kerala leaderName Kochi [/TRIPLE] [/SNT]", *'Output'*: "The AWH Engineering College in Kerala, India was established in 2001. The Ganges is a river in India and its largest city is Mumbai. The leader of Kerala is Kochi.", **5:** *'Input'*: "[SNT] [TRIPLE] Atlanta country United_States [/TRIPLE] [TRIPLE] United_States capital Washington_D.C. [/TRIPLE] [/SNT] [SNT] [TRIPLE] United_States ethnicGroup Asian_Americans [/TRIPLE] [/SNT]", *'Output'*: "Atlanta is in the United States whose capital is Washington, D.C. Asian Americans are an ethnic group in the U.S."
Source	Input triple(s) from the test set. E.g. [SNT] Bananaman broadcastedBy BBC, Bananaman firstAired "1983-10-03", Bananaman lastAired "1986-04-15" [/SNT] [SNT] Bananaman creator John_Geering, Bananaman starring Tim_Brooke-Taylor [/SNT]
Our Prompt(s)	**GPT-4**: {User instruction}\n Examples:{Data examples}\n Input: {Source}\n Output:\n **Mistral7b**: <s>[INST] «SYS» {System instruction}\n{User instruction}\nExamples:{Data examples}«/SYS»\nInput: {source}\nOutput:\n[/INST]

Table 4: Prompt design for English. The first data examples are for direct prompt-based experiments and the latter are for model hybridisation experiments.

Target language	Arabic, Korean, Swahili
Instruction	"Translate the following English language text to {tgt_lang} language text. Provide only the translation. Follow the example below. ######"
Data Examples	**1:** '*Input*': 'Alfredo Zitarrosa died in Montevideo, Uruguay which is led by Raúl Fernando Sendic Rodríguez and Tabaré Vázquez.', '*Arabic*': توفي ألفريدو زيتاروزا في مونتيفيديو، أوروغواي التي يقودها راؤول فرناندو سينديتش رودريغيز وتاباري فاسكيز، '*Korean*': "영국의 수도는 런던으로, 돌로 만든 데드맨스 플랙(Dead Man's Plack)을 찾을 수 있습니다. Plack은 East Anglia의 Ealdorman인 Æthelwald에게 헌정되었습니다. 영국에서는 콘월어가 사용되며 영국 교회라는 종교가 확립되어 있습니다. 그 나라에서 발견되는 인종 그룹 중 하나는 영국계 아랍인입니다.", '*Swahili*': "Mji mkuu wa Uingereza ni London ambapo tunaweza kupata Plack ya Dead Man ambayo imetengenezwa kwa mawe. Plack imejitolea kwa Æthelwald, Ealdorman wa East Anglia. Lugha ya Cornish inazungumzwa nchini Uingereza na ina dini iliyoanzishwa inayoitwa Kanisa la Anglikana. Moja ya makabila yanayopatikana katika nchi hiyo ni Waarabu wa Uingereza.", **2:** '*Input*': 'Angola International Airport is located at Ícolo e Bengo in Luanda province, Angola. The Airport is situated 159 meters above sea level and serves the city of Luanda.', '*Arabic*': يقع مطار أنغولا الدولي في ايكولو ايبينغو في مقاطعة لواندا، أنغولا. يقع المطار على ارتفاع ٩٥١ متراً فوق مستوى سطح البحر ويخدم مدينة لواندا، '*Korean*': "앙골라 국제공항은 앙골라 루안다 지방의 이콜로 에 벤고에 위치해 있습니다. 공항은 해발 159미터에 위치해 있으며 루안다 시에 서비스를 제공합니다.", '*Swahili*': "Uwanja wa ndege wa Kimataifa wa Angola uko Ícolo e Bengo katika jimbo la Luanda, Angola. Uwanja wa ndege upo mita 159 juu ya usawa wa bahari na unahudumia jiji la Luanda.", **3:** '*Input*': 'Akeem Adams, who plays for the Trinidad and Tobago national under-20 football team previously played for United Petrotrin FC whose ground is at Palo Seco.', '*Arabic*': أكيم آدامز، الذي يلعب لصالح منتخب ترينيداد وتوباغو لكرة القدم تحت ٠٢ سنة، سبق له اللعب مع نادي يونايتد بيتروترين لكرة القدم الذي يقع ملعبه في بالو سيكو، '*Korean*': "트리니다드토바고 20세 이하 축구 국가대표팀에서 뛰고 있는 아킴 아담스는 팔로세코를 연고지로 하는 유나이티드 페트로트린 FC에서 선수 생활을 했습니다.", '*Swahili*': "Akeem Adams, anayechezea timu ya taifa ya vijana ya Trinidad na Tobago ya soka ya vijana chini ya umri wa miaka 20 hapo awali aliichezea United Petrotrin FC ambayo uwanja wake ni Palo Seco.", **4:** '*Input*': 'The United States fighter pilot William Anders was born in British Hong Kong on the 17th of October, 1933. In 1963, he was chosen by NASA and became a crew member on Apollo 8.', '*Arabic*': "ولد الطيار المقاتل الأمريكي ويليام أندرس في هونغ كونغ البريطانية في ٧١ أكتوبر ٣٣٩١. وفي عام ٣٦٩١، تم اختياره من قبل وكالة ناسا وأصبح أحد أفراد طاقم أبولو ٨، '*Korean*': "미국 전투기 조종사 윌리엄 앤더스는 1933년 10월 17일 영국령 홍콩에서 태어났어요. 1963년 NASA에 발탁되어 아폴로 8호의 승무원이 되었습니다.", '*Swahili*': "Rubani wa kivita wa Marekani William Anders alizaliwa Uingereza Hong Kong tarehe 17 Oktoba, 1933. Mnamo 1963, alichaguliwa na NASA na kuwa mwanachama wa wafanyakazi kwenye Apollo 8.", **5:** '*Input*': "The capital of England is London where we can find the Dead Man's Plack which is made of stone. The Plack is dedicated to Æthelwald, Ealdorman of East Anglia. Cornish language is spoken in England and it has an established religion called the Church of England. One of the ethnic groups found in that country is the British Arabs.", '*Arabic*': عاصمة إنجلترا هي لندن حيث يمكننا العثور على نصب ديدمان بلاك تذكاري المصنوع من الحجر. المقام مخصص للملك إيثلووف، زعيم و قائد من شرق أنجليا. يتم التحدث باللغة الكورنية في إنجلترا ولها دين راسخ يسمى كنيسة إنجلترا. إحدى المجموعات العرقية الموجودة في ذلك البلد هي العرب البريطانيين، '*Korean*': "영국의 수도 런던에는 돌로 만든 데드맨의 플랙이 있습니다. 이 플랙은 이스트 앵글리아의 에델발드에게 헌정되어 있어요. 영국에서는 콘월어를 사용하며 영국 국교회라는 종교가 확립되어 있습니다. 이 나라에서 발견되는 인종 그룹 중 하나는 영국 아랍인입니다.", '*Swahili*': "Mji mkuu wa Uingereza ni London ambapo tunaweza kupata Plack ya Dead Man ambayo imetengenezwa kwa mawe. Plack imejitolea kwa Æthelwald, Ealdorman wa East Anglia. Lugha ya Cornish inazungumzwa nchini Uingereza na ina dini iliyoanzishwa inayoitwa Kanisa la Anglikana. Moja ya makabila yanayopatikana katika nchi hiyo ni Waarabu wa Uingereza."
Source	System outputs from GPT-4 or Mistral7b. E.g. Aaron Turner, a post-metal singer, started his active years in 1995. He is associated with the band Twilight.
Our Prompt	{instruction} \nExamples:{examples}\nInput: {source}\nOutput:\n

Table 5: Prompt design for translation of English to Arabic, Korean and Swahili using the command-r-plus-4bit model from Cohere AI.

DCU-NLG-PBN at the GEM'24 Data-to-Text Task: Open-Source LLM PEFT-Tuning for Effective Data-to-Text Generation

Michela Lorandi and **Anya Belz**
ADAPT Research Centre, Dublin City University
{michela.lorandi, anya.belz}@adaptcentre.ie

Abstract

LLMs have been used in various tasks with impressive success, including data-to-text generation. However, one concern when LLMs are compared to alternative methods is data contamination, in other words, for many datasets the data used in training these models may have included publicly available test sets. In this paper, we explore the performance of LLMs using newly constructed datasets in the context of data-to-text generation for English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic. We performed a testing phase to evaluate a range of prompt types and a fine-tuning technique on Mistral 7B and Falcon 40B. We then fully evaluated the most promising system for each scenario: (i) LLM prompting in English followed by translation, and (ii) LLM PEFT-tuning in English followed by translation. We find that fine-tuning Mistral outperforms all other tested systems and achieves performance close to GPT-3.5. The few-shot prompting with a dynamic selection of examples achieves higher results among prompting. The human evaluation to be carried out by the shared-task organisers will provide insight into the performance of the new datasets. In conclusion, we observed how the fine-tuning of an open-source LLM can achieve good performance close to state-of-the-art closed-source LLM while using considerably fewer resources.

1 Introduction

With the advancement of Large Language Models (LLMs), their capabilities have been explored in many tasks including data-to-text generation, which maps structured input data into a suitable output text containing all and only provided information. However, the datasets for many data-to-text tasks have been available online for years and might have been used to train LLMs. In the work reported here, we participate in the GEM 2024 shared task (Mille et al., 2024) using new datasets which are not available online.

In more detail, we address the data-to-text generation task using two settings: LLM prompting and fine-tuning. However, fine-tuning LLMs for specific tasks remains challenging, often constrained by computational resources. To mitigate this, we use a Parameter Efficient Fine-Tuning (PEFT) technique to substantially reduce the number of parameters participating in training, making the fine-tuning process far more computationally efficient while maintaining model performance. In both explored settings, we use an external Machine Translation (MT) system to translate our English-generated texts into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

The paper is structured as follows. Section 2 describes data and task, and Section 3 presents the general approach, prompt types, testing phase and the specific systems we fully evaluated. Experimental set-up and results are outlined in Section 4, and Section 5 provides conclusions.

All the code and generated texts are available on GitHub.[1]

2 Data and Task

The Data-to-Text task converts input data, specifically RDF triples representing *subject* | *predicate* | *object* combinations, into coherent and contextually appropriate text that accurately conveys all and only the information present in the input triples.

The GEM 2024 shared task provides datasets for two subtasks: (i) WebNLG-based, utilising the official WebNLG (Castro Ferreira et al., 2020) test set, and (ii) Wikidata-based, using newly obtained triples from Wikidata. Each subtask includes three parallel datasets: Factual, Counterfactual, and Fictional. The Factual dataset consists of triples found in WebNLG or Wikidata. The Counterfactual dataset switches entities based on their class,

[1] https://github.com/michelalorandi/DCU-NLG-PBN-GEM24

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 76–83
September 23 –27, 2024.

creating hypothetical scenarios. Finally, the Fictional dataset replaces original entities with those created via LLM prompting.

For all datasets, only the test set is provided, containing the input triples with predicates in English. No training data is available, and reference texts are not provided. However, for the WebNLG-based Factual dataset, references can be extracted from the original WebNLG English dataset, allowing for some level of automatic evaluation.

3 Systems

We consider two settings to create our systems using pretrained LLMs (Figure 1): (i) generate text in English using out-of-the-box LLMs with prompting, (ii) generate text in English using a fine-tuned LLM. In the first setting, we employ pretrained LLMs without additional training and use various prompting strategies to guide the model in generating text based on the input RDF triples. In the second setting, we fine-tune pretrained LLMs using Low-Rank Adaptation (LoRA). Regardless of the generation method, the generated English text is then translated into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic using a Machine Translation system.

3.1 Prompt types

In our experiments, we used the same prompts proposed by Lorandi and Belz (2023): Zero-shot minimal instruction and Few-shot in-context learning. **Zero-shot minimal instruction** consists of a simple and brief description of the task followed by the input. The prompt does not include any detail or example of the task. **Few-shot in-context learning** contains the same brief task description but adds a list of examples showing both input and target output.

We explored four variations of Few-shot in-context learning, each differing in how examples were selected, based on the idea that choosing examples similar to the input triples would improve the model's performance:

1. *Fixed examples*: The list of examples is fixed for every sample in the dataset.

2. *Dynamic examples based on triple set length*: Examples are randomly selected from the list where the triple set length matches the input triple set length.

3. *Dynamic examples based on properties*: Examples are randomly selected from those that share at least one property with the input; if no such examples exist, a random selection from all examples is performed.

4. *Dynamic examples based on triple set length and properties*: Examples are first filtered by matching triple set length, then randomly selected from those that share at least one property with the input; if no such examples exist, a random selection from the length-matched examples is performed.

3.2 Testing and model selection

We conducted a testing phase using the entire English validation set of WebNLG 2020 to evaluate our settings. We tested two instructed-tuned LLMs for prompting and four LLMs for fine-tuning, resulting in the following experimental grids:

1. {Mistral 7B Instruct, Falcon 40B Instruct} x {Zero Shot, Few Shot, Few Shot with dynamic examples based on triple set length, Few Shot with dynamic examples based on predicates, Few Shot with dynamic examples based on triple set length and predicates} x {English}

2. {Mistral 7B, Mistral 7B Instruct, Falcon 40B, Falcon 40B Instruct} x {WebNLG 2020 (English)} x {LoRA} x {English}

Prompting. We tested all the prompts described in Section 3.1 using Mistral 7B Instruct[2] and Falcon 40B Instruct.[3] For the dynamic selection of examples, we created a pool of possible examples from the train set and translated them into all languages using No Language Left Behind (NLLB) (Costa-jussà et al., 2022). All prompts were tested on the complete validation set of WebNLG 2020. The full text of the used prompts is shown in Appendix A.

Model Fine-Tuning. We PEFT-tuned four different LLMs: Mistral 7B (Jiang et al., 2023), Mistral 7B Instruct, Falcon 40B (Almazrouei et al., 2023), and Falcon 40B Instruct. We used LoRA (Hu et al., 2021) as the PEFT technique to fine-tune the selected models using the training and validation sets

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[3] https://huggingface.co/tiiuae/falcon-40b-instruct

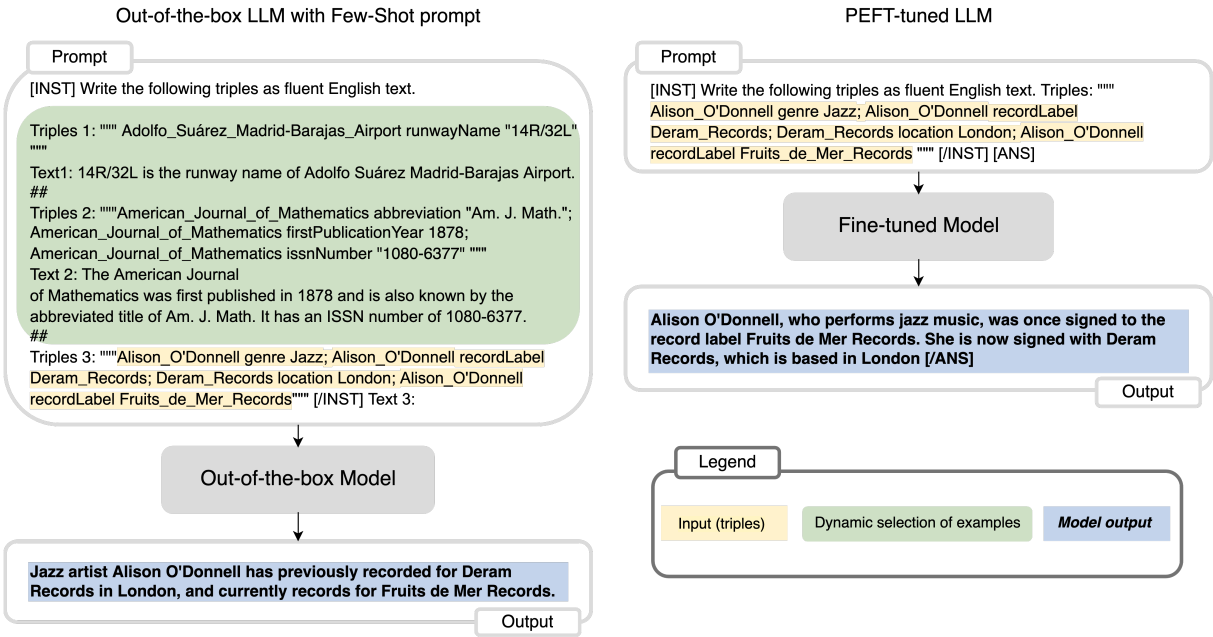


Figure 1: The two systems used in the final evaluation with input and output structure examples. Given Input (triples) highlighted in yellow, model output in blue. The few-shot in-context prompt also incorporates examples (highlighted green).

Model	Setting	BLEU↑	ChrF++↑	TER↓
Mistral 7B	Fine-tuning	**62.878**	**0.75**	**0.33**
Mistral 7B Instruct	Fine-tuning	55.1306	0.71	0.45
	Zero-shot	23.2855	0.58	0.82
	Few-shot fixed	36.8946	0.65	0.61
	Few-shot dynamic, length	36.3098	0.65	0.61
	Few-shot dynamic, properties	38.8017	0.66	0.57
	Few-shot dynamic, length and properties	*40.1638*	*0.67*	*0.55*
Falcon 40B	Fine-tuning	46.0399	0.5	0.55
Falcon 40B Instruct	Fine-tuning	46.0189	0.68	0.48
	Zero-shot	22.0014	0.24	0.82
	Few-shot fixed	25.9916	0.42	0.75
	Few-shot dynamic, length	18.5744	0.21	0.84
	Few-shot dynamic, properties	16.4993	0.17	0.89
	Few-shot dynamic, length and properties	22.2892	0.22	0.81

Table 1: Preliminary automatic evaluation results of our testing phase on the validation set of WebNLG 2020 in English. Best overall system in bold, best prompting system in italics.

of WebNLG 2020. For fine-tuning, we construe the task as an instruction-based task where special tokens delimit the task description, input, and output. The special tokens are designed to train the model to accurately identify the answer, ensuring it includes all and only the information contained in the input, thereby reducing hallucinations and omissions. See Section 4 for more details.

We performed post-processing based on the validation set results in both settings, as follows. We removed special tokens for the start of the sentence, end of the sentence, and padding. The answer was considered to be the text between the special answer tokens in the case of fine-tuning, and the text up to the first occurrence of the character sequence *Triples* or *(Note:* in the case of prompting. We further removed [and] characters and replaced \n with a space.

Table 1 shows the preliminary results from the testing phase. Mistral 7B consistently outperformed alternatives by substantial margins. Furthermore, within the prompting results, Mistral 7B Instruct with Few-Shot prompts using dynamic examples selected based on length and predicates outperformed all other prompting techniques. We selected these configurations as our final systems for submission based on these preliminary results.

Model	Setting	BLEU↑	BLEU NLTK↑	METEOR↑	ChrF++↑	BERT↑ P	R	F1
Mistral 7B Instruct	Fine-tuning	52.26	0.516	0.41	0.679	0.958	0.955	0.956
	Few-shot dynamic	40.12	0.395	0.401	0.655	0.946	0.954	0.949
GPT-3.5 (175B)	Few-shot fixed	**52.74**	**0.519**	**0.417**	**0.69**	**0.959**	**0.958**	**0.958**

Table 2: Automatic evaluation results comparison between our system and Lorandi and Belz (2023) best system (GPT-3.5) on the test set of WebNLG 2020 in English. Best overall system in bold. Few-shot dynamic = Few-shot prompt with dynamic selection of examples based on length and predicates.

System	BLEU↑	METEOR↑	ChrF++↑	BERT-F1↑
DCU-ADAPT-modPB	49.8	0.400	0.655	0.955
DCU-NLG-PBN (our)	**52.26**	**0.410**	0.679	**0.956**
DCU-NLG-Small	51.43	0.395	0.662	0.954
DipInfo-UniTo	51.36	**0.410**	**0.681**	0.955
OSU CompLing	43.09	0.389	0.65	0.950
RDFpyrealb	42.38	0.390	0.642	0.946
SaarLST	39.86	0.400	0.655	0.947

Table 3: Automatic evaluation results on the English test set of WebNLG 2020, comparing the performance of participating systems in the GEM 2024 shared task. Best overall system in bold.

3.3 Prompts and models used in final systems

Based on the results of our testing phase, we evaluated the following system variants as our final systems:

1. {Mistral 7B Instruct} x {Few Shot with dynamic examples based on triple set length and predicates} x {Google Translate} x {English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, Arabic}

2. {Mistral 7B} x {WebNLG 2020 (English)} x {LoRA} x {Google Translate} x {English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, Arabic}

Both systems incorporate the post-processing steps described in Section 3.2. We use Google Translate to translate English-generated texts into Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

4 Experimental Set-up and Results

We executed our experiments using the transformer library[4] of HuggingFace and the paid-for Google Translate API[5] in late March/early April 2024. The systems are tested using the six datasets described in Section 2. All generated texts are post-processed as described in Section 3.2. All systems are executed on a Nvidia A100 GPU with 80GB RAM.

Prompting. We set mistralai/Mistral-7B-Instruct-v0.2 parameters to *max seq length*=512, *seed*=6787, and *use 4bit*=True.

Model Fine-tuning. We use the PEFT library[6] of HuggingFace to create and load LoRA modules. We set mistralai/Mistral-7B-v0.1 parameters to *max steps*=10000, *learning rate*=2e-4, *max grad norm*=0.3, *weight decay*=0.001, *lora alpha*=16, *lora dropout*=0.1, *lora r*=64, *max seq length*=512, *seed*=6787, *use 4bit*=True, and *warmup ratio*=0.3. We use the checkpoint at step 6000 at inference time as it has the lowest loss based on the validation set. WebNLG 2020 train set is used for the model fine-tuning. The fine-tuning is defined as an instruction-based task where the task description and input are delimited by special instruction tokens ([INST] and [/INST]), and the output is delimited by special answer tokens ([ANS] and [/ANS]).

Following the WebNLG 2023 evaluation setup (Cripwell et al., 2023), we perform an automatic evaluation on the WebNLG-based Factual dataset in English computing BLEU (Papineni et al., 2002), ChrF++ (Popović, 2017), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al.). We compare our two systems against the best system proposed by Lorandi and Belz (2023), i.e. GPT-3.5 using Few-Shot prompt with fixed examples.

An additional human evaluation will be performed by the organisers of the shared task and

[4] https://huggingface.co/docs/transformers/index

[5] https://cloud.google.com/translate

[6] https://huggingface.co/docs/peft/index

at the time of writing the results are not available yet. Refer to the shared task report for more details.

Table 2 shows the results of the automatic evaluation in English on the WebNLG-based Factual dataset, for which references are available. Our fine-tuned model outperforms the prompting-based Mistral 7B Instruct by clear margins. Scores for GPT-3.5 are higher than for fine-tuned Mistral 7B Instruct by tiny margins in all cases. However, the latter achieves these very close results while utilising a substantially smaller model size (25x). This significant reduction in model size translates to lower computational costs, decreased memory usage, and faster processing times, making the fine-tuned Mistral 7B a more resource-efficient option.

Table 3 shows the automatic evaluation results in English on the WebNLG-based Factual dataset comparing all participating systems in the GEM 2024 shared task. Our fine-tuned system (DCU-NLG-PBN) shows strong performance, achieving the highest scores in both BLEU and BERT-F1. DipInfo-UniTo system, while slightly lower in BLEU, leads in ChrF++ and performs competitively in METEOR, alongside our system. These results, however, represent partial evaluations on the WebNLG-based Factual dataset using all available references. More insights on the performance of the systems will emerge from the human evaluation results. For additional automatic evaluation results, refer to the shared task report (Mille et al., 2024).

5 Conclusion

We explored the effectiveness of pretrained LLMs for data-to-text generation focusing on two settings: LLM prompting and LLM fine-tuning with LoRA. We first conducted a testing phase comparing the performance of Mistral 7B and Falcon 40B models using various prompting strategies and fine-tuning techniques, evaluated on the WebNLG 2020 validation set. The results demonstrated that fine-tuning with LoRA substantially enhances the performance of the Mistral 7B model. This model outperformed all other tested systems, including Falcon 40B. Among the prompting strategies, the few-shot in-context learning with dynamic examples based on the triple set length and predicates achieved the best results, indicating the importance of contextually relevant example selection.

We submitted the two system settings, LLM prompting + MT and LLM fine-tuning with LoRA + MT, using Mistral 7B to the GEM 2024 shared task in English, Chinese, German, Russian, Spanish, Korean, Hindi, Swahili, and Arabic.

Our findings highlight the potential of LoRA for efficient fine-tuning of LLMs, offering a competitive performance close to state-of-the-art models like GPT-3.5, but with substantially smaller model sizes and reduced resource requirements. The success of dynamic example selection in prompting also underscores the need for tailored approaches to optimize model performance.

Acknowledgments

Michela Lorandi's work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. Both authors benefit from being members of the ADAPT SFI Research Centre at Dublin City University, funded by the Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Liam Cripwell, , Anya Belz, Claudia Borg, Claire Gardent, Albert Gatt, John Judge, Michela Lorandi, Anna Nikiforoskaya, William Soto-Martinez, and Craig Thomson. 2023. The 2023 webnlg shared task on low resource languages overview and evaluation results (webnlg 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge*, Prague, Czech Republic.

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

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. *arXiv preprint arXiv:2310.06825*.

Michela Lorandi and Anya Belz. 2023. Data-to-text generation for severely under-resourced languages with GPT-3.5: A bit of help needed from Google Translate (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 80–86, Prague, Czech Republic. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. 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.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Prompts

Table 4 shows the input template of our fine-tuned Mistral 7B Instruct. Tables 5 and 6 shows the Zero-shot prompts used with Mistral 7B Instruct and Falcon 40B Instruct, while Tables 7 and 8 shows the Few-shot prompts used with Mistral 7B Instruct and Falcon 40B Instruct.

Input template fine-tuned Mistral 7B	
Template:	[INST] Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] [ANS] {MODEL} [/ANS]
Example:	[INST] Write the following triples as fluent English text. Triples: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] [ANS]
Model output:	*Alison O'Donnell, who performs jazz music, was once signed to the record label Fruits de Mer Records. She is now signed with Deram Records, which is based in London [/ANS]*

Table 4: Input template for our fine-tuned Mistral 7B. **Top** Template of the input. **Bottom** Complete example in English.

Zero-Shot prompt Mistral 7B Instruct	
Template:	[INST] Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] Text: {MODEL}
Example:	[INST] Write the following triples as fluent English text. Triples: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] Text:
Model output:	*Alison O'Donnell is a jazz artist. She is signed under the record label Deram Records. Deram Records is based in London. Alternatively, Alison O'Donnell has also recorded under the record label Fruits de Mer Records.*

Table 5: Zero-shot prompt for Mistral 7B Instruct. **Top** Template of the input. **Bottom** Complete example in English.

Zero-Shot prompt Falcon 40B Instruct	
Template:	»QUESTION« Write the following triples as fluent English text. Triples: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ »ANSWER« Text: {MODEL}

Table 6: Zero-shot prompt for Falcon 40B Instruct.

Few-Shot prompt Mistral 7B Instruct	
Template:	[INST] Write the following triples as fluent English text. Triple 1: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 1: {verbalisation of Triple 1} ## Triple 2: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 2: {verbalisation of Triple 2} ## Triple 3: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ [/INST] Text 3: {MODEL}
Fixed examples:	Triple set 1: Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. Triple set 2: American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377.
Example Prompt:	[INST] Write the following triples as fluent English text. Triple 1: """ Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" """ Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. ## Triple 2: """ American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" """ Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377. ## Triple 3: """ Alison_O'Donnell genre Jazz; Alison_O'Donnell recordLabel Deram_Records; Deram_Records location London; Alison_O'Donnell recordLabel Fruits_de_Mer_Records """ [/INST] Text 3:
Model output:	*Jazz artist Alison O'Donnell has previously recorded for Deram Records in London, and currently records for Fruits de Mer Records.*

Table 7: Few-Shot prompt for Mistral 7B Instruct. **Top** Template of the prompt. **Middle** Fixed examples used during testing. **Bottom** Complete example in English.

Few-Shot prompt Falcon 40B Instruct	
Template:	»QUESTION« Write the following triples as fluent English text. Triple 1: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 1: {verbalisation of Triple 1} ## Triple 2: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ Text 2: {verbalisation of Triple 2} ## Triple 3: """ {set of triples in the format *subject predicate object* and each triple separated by ;} """ »ANSWER« Text 3: {MODEL}
Fixed examples:	Triple set 1: Adolfo_Suárez_Madrid–Barajas_Airport runwayName "14R/32L" Text 1: 14R/32L is the runway name of Adolfo Suárez Madrid-Barajas Airport. Triple set 2: American_Journal_of_Mathematics abbreviation "Am. J. Math."; American_Journal_of_Mathematics firstPublicationYear 1878; American_Journal_of_Mathematics issnNumber "1080-6377" Text 2: The American Journal of Mathematics was first published in 1878 and is also known by the abbreviated title of Am. J. Math. It has an ISSN number of 1080-6377.

Table 8: Few-Shot prompt for Falcon 40B Instruct. **Top** Template of the prompt. **Bottom** Fixed examples used during testing.

DCU-NLG-Small at the GEM'24 Data-to-Text Task: Rule-based generation and post-processing with T5-Base

Simon Mille, Mohammed Sabry and Anya Belz
ADAPT, Dublin City University
firstname.lastname@adaptcentre.ie

Abstract

Our submission to the GEM data-to-text shared task aims to assess the quality of texts produced by the combination of a rule-based system with a language model of reduced size. Our system first uses a rule-based generator to convert input triples into semantically correct English text, and then a language model to paraphrase these texts to make them more fluent. The texts are translated to languages other than English with the NLLB machine translation system.[1]

1 Introduction

On the one hand, Very Large Language Models are able to produce human-like texts from structured data but require enormous amounts of energy and computational resources to be trained, fine-tuned and run; on the other hand, resource-efficient techniques such as rule-based systems generally output texts that are less than optimally fluent. For our submission, we used three components: (i) a rule-based generator, FORGe (Mille et al., 2023b) to generate all inputs in English, (ii) a small-sized language model, T5-Base (Raffel et al., 2020), fine-tuned for rephrasing the rule-based outputs in a more fluent way, and (iii) an off-the-shelf Machine Translation system, NLLB (Team et al., 2022), for producing outputs in languages other than English. Our hypothesis is that using a language model for paraphrasing textual output produced by a reliable rule-based generator, rather than for directly mapping from triples to text, will make the system (i) more accurate in term of contents, i.e. less prone to omissions and additions (since all the contents of the input triples are already verbalised in the input of the language model), and (ii) generalise better to out-of-domain data, which represents five out of the six test sets of the GEM D2T task (since for the language model, instead of verbalising, the task is now paraphrasing, for which much more training data is available).

[1] Our code and data is available at https://github.com/dcu-nlg/GEM24-DCU-NLG-Small.

Input:

Subject	Property	Object
The_Haunted_Castle	imdbId	12
The_Haunted_Castle	director	Ezekiel_Kemboi
The_Haunted_Castle	director	Oleksandr_Turchynov

Possible English output:
Ezekiel Kemboi and Oleksandr Turchynov are the directors of The Haunted Castle, which has the IMDb identifier "12".

Figure 1: Sample GEM counterfactual input/output pair (D2T-1-CFA dataset).

In the remainder of the paper, we briefly summarise the GEM D2T shared task (Section 2), the rule-based generator and its extension (Section 3), the datasets we collected for fine-tuning T5 (Section 4), the fine-tuning procedure (Section 5), and the use of machine translation (Section 6); finally, we comment on the preliminary results (Section 7).

2 The GEM D2T Shared Task

In GEM D2T (Mille et al., 2024), the task is to generate texts in various languages starting from input triples extracted from DBpedia (Subtask 1) or Wikidata (Subtask 2) triples; see Figure 1 for an example of an input/output pair. Each subtask has three test sets: (i) a factual dataset (FA), which contains only factually correct information; (ii) a counterfactual dataset (CFA), which is the factual dataset but with entities (Subjects and Objects, see Figure 1) replaced by other entities of the same category (e.g. a person is replaced by another person, a date by another date, etc.); and (iii) a fictional dataset (FI), in which all Subject and Object values are fictional names made up by a language model.

The D2T-1 data is derived from WebNLG data (Castro Ferreira et al., 2020), while the D2T-

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 84–91
September 23 –27, 2024. ©2024 Association for Computational Linguistics

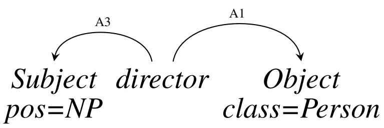


Figure 2: Sample PredArg template corresponding to the *director* property.

2 data was created for the present task using the method proposed by Axelsson and Skantze (2023) (i.e. collection of new Wikidata triples sets for a list of entities, and then replacement of entities according to steps (ii) and (iii) above). No training data was provided to the participants, and apart from the English Factual WebNLG data (i.e. the original test set in (Castro Ferreira et al., 2020)), no reference texts were available for any test set or language. The GEM organisers encouraged submissions in multiple languages, namely English (en), Chinese (zh), German (de), Russian (ru), Spanish (es), Korean (ko), Hindi (hi), Swahili (sw), and Arabic (ar), without saying beforehand which languages were going to be assessed.

3 Rule-based Generator

For our rule-based system, we use the FORGe generator (Mille et al., 2023b), which was partly developed on the WebNLG data. FORGe is implemented as a pipeline of modules that perform subtasks such as text planning, lexicalisation, sentence structuring and surface realisation. Each module consists of a set of rules (called *grammars*), which use dictionaries that describe the semantic and syntactic behaviours of the lexical units used in the verbalisations. The generator takes as input abstract predicate-argument structures manually crafted for each property, as shown in Figure 2.

FORGe already has such predicate-argument structures for the whole WebNLG 2020 dataset in English, which means that we were able to use FORGe off-the-shelf for Subtask 1; no modification was performed to address new entity names of the fictional test set. For Subtask 2, properties in the dataset built by the organisers come from the Wikidata vocabulary, which is different from the DBpedia vocabulary used in the WebNLG dataset. There are 74 different Wikidata properties, 17 of which have a direct mapping to a DBpedia property. For these 17 properties, we use the existing predicate-argument templates, while for the remaining 57 properties, new predicate-argument templates were crafted, referring to the Wikipedia pages of the entities used along each property to make sure we captured the correct semantics of each property. Crafting the 57 templates took approximately 2 hours. Minor updates to the generator's grammars were implemented to account for the specific aspects of the Wikidata test sets, in which the Subject is always the same, unlike in the WebNLG-based inputs.

4 Finetuning Datasets

Our objective in the paraphrasing component is to improve the fluency of the rule-based generator without sacrificing its semantic accuracy (i.e. avoiding what is commonly reported as omissions and hallucinations). For this, we collected parallel textual data, with on one side accurate but possibly disfluent texts (Text_{Dis}), and on the other side accurate and fluent texts (Text_{Flu}). In this section, we describe the three different datasets we created for the experiments; Section 5 reports on how we used this data for fine-tuning T5.

4.1 The forge2ref dataset

For data of type Text_{Dis}, we used texts generated with the FORGe rule-based system (see Section 3) as provided in the English version of the Mod-D2T dataset (Mille et al., 2023a),[2] which is a 10-layer version of the whole WebNLG 2020 dataset (training, development and test sets) produced with FORGe. For the parallel data of type Text_{Flu}, we used the corresponding list of reference texts from the original WebNLG 2020 data in each case, downloaded from HuggingFace.[3] The final data contains 13,211, 1,667, and 1,779 pairs in the training, development and test sets, respectively. The following is an example pair:

- Text_{Dis}: *The production of the Pontiac Rageous started in 1997. The Pontiac Rageous is a coupe.*
- Text_{Flu}: *['The Pontiac Rageous coupe went into production in 1997.', 'The Pontiac Rageous, first produced in 1997, was a car with a coupe body style.', 'The coupe style Pontiac Rageous was first produced in 1997.']*

4.2 The forge2llm dataset

In order to acquire additional high quality data, we also collected a very small set of language model

[2] https://github.com/mille-s/Mod-D2T/tree/main/conllu-en_INLG23

[3] https://huggingface.co/datasets/webnlg-challenge/web_nlg

outputs, using the best systems and the human evaluation results of the WebNLG 2020 shared task. Three systems competing in the 2020 edition of the shared task achieved human-level fluency: AmazonAI (Guo et al., 2020), FBConvAI (Yang et al., 2020) and OSU Neural NLG (Li et al., 2020). Assuming that these systems are generally able to output very fluent text, we selected the subset of these system outputs that were rated 0.95 or more when computing the mean for the three criteria related with the semantic faithfulness to the input triples, namely:

- "DataCoverage: *Does the text include descriptions of all predicates presented in the data?*;
- Relevance: *Does the text describe only such predicates (with related subjects and objects), which are found in the data?*;
- Correctness: *When describing predicates which are found in the data, does the text mention correct the objects and adequately introduces the subject for this specific predicate?*" (*sic*).

The system outputs and human ratings were obtained from the WebNLG GitHub repository.[4] For 163 inputs, we found between one and three system outputs that met our threshold (301 texts in total). These 163 lists of texts served as Text_{Flu} data, and were paired with the corresponding FORGe texts serving as Text_{Dis}, e.g.:

- Text_{Dis} (same as forge2ref's Text_{Dis}): *The production of the Pontiac Rageous started in 1997. The Pontiac Rageous is a coupe.*
- Text_{Flu}: *['The Pontiac Rageous has a Coupe body style and its production started in 1997.', 'Production of the Pontiac Rageous Coupe began in 1997.']*

Note that the data we are using for the forge2llm dataset constitutes about 9% of the D2T-1-FA test set (we use 163 data points out of the 1,779 data points in the test set). We thus expect this to slightly inflate our metrics scores on the D2T-1-FA set, but should not have an important impact on the other test sets.

4.3 The triple2ref dataset

For this dataset, we paired triples and human-written texts, both extracted from the WebNLG 2020 dataset (Castro Ferreira et al., 2020). The input triples are simply concatenated with a comma and a space, and the output reference texts are combined into a list. This dataset is used in addition to the other two for one of the models in order to increase its robustness to bad inputs. The final data contains 13,211, 1,667, and 1,779 pairs in the training, development and test sets respectively, e.g.:

- Text_{Dis}: *Pontiac_Rageous | productionStartYear | 1997, Pontiac_Rageous | bodyStyle | Coupe*
- Text_{Flu} (same as forge2ref's Text_{Flu}): *['The Pontiac Rageous coupe went into production in 1997.', 'The Pontiac Rageous, first produced in 1997, was a car with a coupe body style.', 'The coupe style Pontiac Rageous was first produced in 1997.']*

5 Paraphrasing with T5-Base

In this section, we introduce T5-Base and all the models fine-tuned for our experiments.

5.1 T5: Experimental setup and model configuration

We conducted experiments with the T5-Base V1 model (250M parameters),[5] alongside one full-tuning technique and two parameter-efficient fine-tuning (PEFT) techniques, namely LoRA (Hu et al., 2021) and Adapter (Houlsby et al., 2019). The primary task was text-to-text generation, with the aim of transforming FORGe outputs into more fluent text. The T5-Base model does not inherently possess task-specific knowledge relevant to this task, but it is well-suited for text-to-text modelling tasks like paraphrasing.

For the evaluation phase, the model generation settings were as follows: Temperature: 0.1; Top-k: 100; Top-p: 0.95; Repetition penalty: 0.8.

5.2 Fine-tuning experiments

All models were trained using cross-entropy loss. For evaluating performance, we employed the HuggingFace Evaluate Library[6] to calculate the BLEU[7] and METEOR[8] metrics, comparing the predicted text against all available references for each input.

[4] https://github.com/WebNLG/challenge-2020

[5] https://huggingface.co/google/t5-v1_1-base

[6] https://huggingface.co/docs/evaluate/en/index

[7] https://huggingface.co/spaces/evaluate-metric/bleu

[8] https://huggingface.co/spaces/evaluate-metric/meteor

We used the training and development sets, keeping the test set for final model evaluation.

In our experiments, we tested three different fine-tuning techniques:

- **Full-tune** involves updating all parameters of a model to better suit a downstream task. This traditional method, while effective, becomes increasingly costly as model sizes scale up, prompting research into more parameter-efficient alternatives (Sabry and Belz, 2023).
- **Adapter** (Houlsby et al., 2019) is a parameter-efficient fine-tuning technique where a small set of trainable parameters, typically two linear layers with an activation function in between, is inserted at strategic locations within a model, such as after the attention and feed-forward blocks of a transformer model. Only these newly introduced parameters are updated during finetuning, while the original model's parameters remain fixed. We implemented Adapter with a bottleneck dimension of 64 and a GeLU activation function.
- **LoRA** (Hu et al., 2021) adopts a similar approach, adding a small set of trainable parameters; however, it specifically targets the query and key matrices within the attention blocks of transformers. These added parameters are viewed as a reparameterised form of the existing matrices, designed to accommodate task-specific adjustments without altering the original, fixed parameters of the model. We used LoRA Configuration of a rank of 8, alpha of 16, and a dropout rate of 0.0.

For each fine-tuning technique, we tested 4 sets of conditions involving different combinations of datasets from Section 4 to assess their performance, for a total of 12 different fine-tuned models:

- **10K**: Fine-tuning for 10,000 learning steps exclusively on forge2ref;
- **15K**: Fine-tuning for 15,000 learning steps on the two datasets that use FORGe texts, forge2ref and forge2llm;
- **35K**: Fine-tuning for 35,000 learning steps on all three datasets: forge2ref, forge2llm and triple2ref;
- **Avg.Prm**: Average of the trainable parameters from each of the 3 fine-tuning techniques (Full-tune, Adapters, LoRA, see above). This approach is based on findings that averaging multiple checkpoints can lead to better generalisation (Izmailov et al., 2018).

All models were trained using a learning rate of 6e-5, with a Cosine decay scheduler and 10% of the learning steps designated as a warm-up period. The training and evaluation batch sizes were set at 16.3K and 4K tokens, respectively. Additionally, a weight decay of 0.1 was implemented.

Training the T5-Base in NVIDIA A100-SXM-80GB for 10,000 steps with full precision (FP32), in our initial experiment, required the following GPU durations: 1 hour and 14 minutes for full fine-tuning, 55 minutes for Adapters, and 50 minutes for LoRA. This resulted in total computations of 86 petaFLOPs for full fine-tuning, 64 petaFLOPs for Adapters, and 58 petaFLOPs for LoRA, with corresponding energy consumptions of 0.493 kWh (kilowatt-hours), 0.367 kWh, and 0.333 kWh, respectively. Scaling the same settings to T5-Large could require roughly 3.5 times more, considering the difference in the parameters of the two models (0.2B for T5-Base vs. 0.7B for T5-Large).

When running the paraphrasing, each input is encoded in a maximum of 512 tokens, and the model is set to generate a maximum of 400 tokens. With an A100 GPU, T5 base (FP32) can process about 9K tokens per seconds, which means the paraphrasing time for one full test set (1,8K texts) is about 3 minutes.

6 Machine Translation with NLLB

The combination of resources we need for applying our approach (rule-based generator + parallel textual data) is currently only available in English. For producing outputs in other languages, we used the freely available NLLB machine translation tool (Team et al., 2022). NLLB is a pre-trained model that covers translation between numerous languages; it is available through HuggingFace[9] and can be executed on various types of runtimes, including CPUs. Each English text was split into sentences, and sentences were processed one at a time by NLLB; the translated sentences were then brought back together as a text and stored in the same format as the English outputs, in a .txt file with one text per line. We ran nllb-200-distilled-1.3B on a T4 GPU on Google Colab, which generally needed between 30 and 60 minutes to translate

[9] https://huggingface.co/docs/transformers/en/model_doc/nllb

Fine-tuning	Cond.	BLEU	METEOR
LoRA	10K	0.251	0.494
	15K	0.295	0.536
	35K	0.373	0.585
	Avg.Prm	0.305	0.544
Adapters	10K	0.480	0.670
	15K	0.476	0.671
	35K	0.508	0.694
	Avg.Prm	0.487	0.682
Full-tune	10K	0.506	0.702
	15K	0.542	0.721
	35K	0.536	0.719
	Avg.Prm	0.538	0.719

Table 1: BLEU and METEOR scores (with multiple references) of the 12 fine-tuned T5-Base models on the D2T-1-FA test set; see Section 5.2 for details about the fine-tuning techniques and the conditions.

a file. With 48 files to translate (8 target languages, 6 test sets per language) and several server interruptions, we had to finish the translations on a local HPC cluster to finish the translations on time.

We did not try to improve the translation quality, and did not perform any systematic qualitative analysis of the translated texts; for a few languages (Spanish, Hindi, German), we asked native speakers to browse through a few translations to have an idea of the general quality, which was judged sufficient to submit the outputs.

7 Results and Submitted Systems

In this section, we present the results of evaluating our 12 models on the D2T-1-FA test data, using the WebNLG 2020 reference texts for calculating BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), based on which we selected the model for our submission. We then briefly discuss the official results of the GEM D2T task as provided by the organisers, which at this point are restricted to the metrics scores for English (we report BLEU, METEOR and BertScore-F1 (Zhang et al., 2019)).

7.1 Own evaluation of the fine-tuned models

In Table 1, we report our own BLEU and METEOR scores on the English factual dataset of Subtask 1 (D2T-1-FA), the only one for which references are available at the time of writing.

For all systems, both metrics indicate the same tendencies: the **Full-tune** technique produces the highest scores, closely followed by **Adapter**. With **LoRA**, the results are much lower. We attribute this performance to the small set of parameters added by LoRA, the fact that they interact with the Attention block queries and keys, whereas the tasks require extensive manipulation of factual knowledge, stored in and retrieved from the FeedForward block (Geva et al., 2021). However, increasing the model size, carefully selecting hyperparameters, and/or extending the number of learning steps could mitigate these issues.

With **LoRA**, the more data, the better the results, while that is not necessarily the case for the other two techniques: **Adapter** produces very similar scores with (15K) and without (10K) the forge2llm data but gets better when adding the triple2ref data and learning steps (35K). **Full-tune** benefits more from the forge2llm data (15K) but not from adding the learning steps and the triple2ref data (35K). Since full fine-tuning involves adjusting a larger number of parameters, which allows for a greater degree of freedom to change, the model may initially focus on noisy signals before achieving convergence or being steered in the desired direction by the introduction of triple2ref data (intended to enhance model robustness). We suspect that the number of learning steps allocated may not be sufficient to accommodate these changes.

Finally, averaging the weights from the three fine-tuning techniques produces scores that are between those obtained for 15K and 35K learning steps, in terms of both BLEU and METEOR.

7.2 Submissions

We submitted the *Full-tune Avg.Prm* model, which did not obtain the absolute highest scores for both metrics, but which is supposed to be more robust to input variations (**Ours** in Table 2). As a secondary

System ID	BLEU	METEOR	Bert F1
System 2	52.26	0.410	0.956
Ours	**51.43**	**0.395**	**0.954**
System 4	51.36	0.410	0.955
System 1	49.8	0.400	0.955
System 5	43.09	0.389	0.950
System 6	42.38	0.390	0.946
Ours$_{NoT5}$	**40.55**	**0.372**	**0.943**
System 7	39.86	0.400	0.947
System 8	34.71	0.280	0.923

Table 2: Metrics evaluation of our *Full-tune Avg.Prm* system on the WebNLG 2020 test set provided by the organisers (sorted by BLEU score).

	D2T-1			D2T-2		
	FA	**CFA**	**FI**	**FA**	**CFA**	**FI**
BLEU	27.0	22.98	20.85	19.48	24.9	16.88
METEOR	0.314	0.279	0.292	0.26	0.3	0.267
chrF++	0.537	0.488	0.507	0.438	0.51	0.442
BERT F1	0.93	0.918	0.914	0.925	0.923	0.914

Table 3: Metrics scores for our DCU-NLG-Small submission for the English D2T task released by the organisers.

submission, and for comparison, we submitted all outputs of the rule-based generator without the T5 post-processing (**Ours**$_{NoT5}$ in Table 2). We also submitted outputs for all languages other than English, all produced by running NLLB off-the-shelf on the FORGe+T5 outputs.

7.3 GEM automatic evaluation results

Table 2 shows the first results released by the organisers, i.e. the metrics for the full English test set using all WebNLG 2020 reference texts (1,779 inputs, 2.5 reference texts per input on average). The scores of our system on this dataset cannot be clearly interpreted, since as mentioned in Section 4.2, we use a small portion of this dataset to fine-tune one of the models whose parameters were averaged to make the submitted model. One thing that can be noticed is the extent of the increase of the BLEU score when integrating the T5 post-processing. With close to 11 BLEU points difference, this suggests that our system outputs with T5 are much more similar to the reference texts than the FORGe outputs, which was expected.[10]

The organisers then released metrics results on the 6 D2T test sets (180 inputs each), using references collected on Amazon Mechanical Turk (one reference text per input); see Table 3. On the D2T-1 datasets, our system's scores substantially drop on the counterfactual (CFA) and fictional (FI) datasets; compared to the other participating systems, ours actually is the one that has the most substantial score decrease. In contrast, we have one of the smallest decreases between the factual (FA) and fictional (FI) D2T-2 datasets. Surprisingly, the D2T-2-CFA scores are higher than the D2T-1-CFA counterpart, and also than the D2T-2-FA score. However, all participating systems exhibited the same patterns, so it is likely that the data is somewhat responsible for this oddity. In general, the results of the human evaluation on the 6 test sets will shed more light on the actual quality of the contents produced by our system.

8 Conclusions

We have presented the DCU-NLG-Small submission to the GEM'24 Data-to-text shared task. Our system combines a rule-based generator that converts triples into English text, with a small language model that paraphrases the text to improve its fluency. An off-the-shelf MT system is used for producing outputs in the other languages. Our system performs better than a purely rule-based system according to metrics on an existing English test set, but generally undergoes substantial score decreases when confronted with different types of out-of-domain data. We hope that the human evaluation results will allow us to draw more definitive conclusions.

Acknowledgements

We would like to thank Michela Lorandi for her help with running the code to produce the machine-translated outputs. Mille's work was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS). Sabry's PhD is funded by the ADAPT SFI Centre for Digital Media Technology. Our work on this paper has also benefited more generally from being carried out within the ADAPT SFI Centre which is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

References

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language

[10] Regarding the differences between the scores in Tables 1 and 2: for BLEU, there are differences between the evaluation package we used (Evaluate library from HuggingFace) and the commonly used WebNLG evaluation package, in particular in the smoothing factors applied in the BLEU metric calculation. This explains the 2.5-point discrepancy in BLEU scores observed between the results labelled 'Avg. Prm Full-tune' in Table 1 and 'Ours' in Table 2. In addition, for our own computation of METEOR, we used multiple references, as opposed to single references for the organisers, so the METEOR scores in Tables 1 and 2 are not comparable.

generation from knowledge graphs. *arXiv preprint arXiv:2307.07312*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qipeng Guo, Zhijing Jin, Ning Dai, Xipeng Qiu, Xiangyang Xue, David Wipf, and Zheng Zhang. 2020. $\mathcal{P}^2$: A plan-and-pretrain approach for knowledge graph-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 100–106, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *CoRR*, abs/1803.05407.

Xintong Li, Aleksandre Maskharashvili, Symon Jory Stevens-Guille, and Michael White. 2020. Leveraging large pretrained models for WebNLG 2020. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 117–124, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Simon Mille, Francois Lareau, Stamatia Dasiopoulou, and Anya Belz. 2023a. Mod-D2T: A multi-layer dataset for modular data-to-text generation. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 455–466, Prague, Czechia. Association for Computational Linguistics.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, Elaine Uí Dhonnchadha, Lauren Cassidy, Brian Davis, Stamatia Dasiopoulou, and Anya Belz. 2023b. 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.

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.

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.

Mohammed Sabry and Anya Belz. 2023. Peft-ref: A modular reference architecture and typology for parameter-efficient finetuning techniques. *Preprint*, arXiv:2304.12410.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. 2020. Improving text-to-text pre-trained models for the graph-to-text task. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 107–116, Dublin, Ireland (Virtual). Association for Computational Linguistics.

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

TeamSaarLST at the GEM'24 Data-to-text Task: Revisiting symbolic retrieval in the LLM-age

Mayank Jobanputra and **Vera Demberg**
{firstname}@lst.uni-saarland.de
Department of Language Science and Technology
Saarland University

Abstract

Data-to-text (D2T) generation is a natural language generation (NLG) task in which a system describes structured data in natural language. Generating natural language verbalization for structured data is challenging as the data may not contain all the required details (here, properties such as gender are missing from the input data and need to be inferred for correct language generation), and because the structured data may conflict with the knowledge contained in the LLM's parameters learned during pretraining. Both of these factors (incorrect filling in of details, pretraining conflict and input data) can lead to so-called hallucinations.

In this paper, we propose a few-shot retrieval augmented generation (RAG) system, using a symbolic retriever – `PropertyRetriever`. Additionally, we experiment with state-of-the-art large language models (LLMs) to generate data verbalizations. Our system achieves the best results on 4 out of 6 subtasks for `METEOR` and `chrF++` metrics. We present our results along with an error analysis. We release our code for reproducing the results as well as the generated verbalizations from all the experiments for any further explorations here.[1]

1 Introduction

Nowadays LLMs are pretrained using trillions of text tokens[2] (Penedo et al., 2024). These LLMs can not only generate grammatical and fluent text, but they are also capable of learning new tasks without any training data using in-context learning techniques (Lampinen et al., 2022). One central challenge in LLMs research is to understand the extent to which LLMs memorize their training data versus how they generalize to new tasks and settings. There has been some empirical evidence that LLMs do some degree of both: they clearly memorize parts of the training data – for example, LLMs are often able to reproduce large portions of training data verbatim (Yu et al., 2023; Carlini et al., 2023) – but LLMs also seem to learn from this data, allowing them to generalize to new tasks. Do LLMs truly produce new content, or do they only remix their training data? Until we concretely answer this question, it is essential to test model faithfulness systematically through various data augmentation techniques.

The task of data-to-text generation is one of the popular NLG tasks. In this task, the system is given a set of RDF triplets describing facts (i.e., entities and relations between them) and has to produce a fluent text that is faithful to the facts. The GEM'24 (Mille et al., 2024) challenge brings forth a new shared task on data-to-text generation to test LLMs for factual information (i.e., information in the model parameters is likely to be in line with the input), vs. counterfactual information (i.e., the information in the prompt contrasts with what the model encodes about this entity) vs. fictional entities (i.e., the model parameters should not contain specific information supporting or contradicting the prompt information.)

The GEM'24 shared task consists of two subtasks of generating texts from input triple sets (*Subject* | *Property* | *Object*) in the WebNLG fashion. We participate in both the subtasks. One of the subtasks (D2T-1) is based on the WebNLG dataset. This subtask uses the official WebNLG test set[3] as input for testing the generation system. The test data contains 1,779 input triples with properties and entities not seen in the training/dev data. The second subtask (D2T-2) is based on the Wikidata. This subtask uses 1,800 newly compiled input triples from Wikidata for testing the generation system. Axelsson and Skantze (2023) proposed this dataset containing 74 new properties and entities,

[1] https://github.com/mayankjobanputra/d2t-gem
[2] https://www.together.ai/blog/redpajama-data-v2
[3] https://huggingface.co/datasets/GEM/web_nlg

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 92–99
September 23 –27, 2024.

which were not part of the WebNLG dataset.

In recent years, LLMs such as GPT-4 (Achiam et al., 2023), Gemini (Team et al., 2023) and LLaMa (Touvron et al., 2023), have made significant advancements in the field of natural language generation (NLG). However, the inherent tendency of these LLMs to generate inaccurate or non-factual content, commonly referred to as "hallucinations" (Puzikov and Gurevych, 2018; Ji et al., 2023), continues to present a significant challenge. This generally occurs because the model parameters from pretraining encode some information, which may "overwrite" the information in the prompt due to its high sequence probability. Another challenge with structured data is that the data does not contain all the required details such as entity type, gender and relation explanation. If the model fails to infer these details correctly, it may generate hallucinated verbalization.

In the literature, Shuster et al. (2021) suggests that providing relevant examples during inference can help in reducing hallucinations. While Moryossef et al. (2019) suggests using an explicit, symbolic, text planning stage for generating more faithful verbalization of data. In this work, we combine these suggestions and propose a few-shot RAG system to solve this task, using a symbolic retriever - `PropertyRetriever`. Figure 1 illustrates the architecture of our system. We experiment with state-of-the-art open-weight models for generating verbalization. In the following sections, we describe the dataset, our approach and provide a detailed study of the errors made by the system.

2 Dataset

The GEM'24 shared task introduced novel augmented test sets for both WebNLG and Wikidata. These augmented test sets consist of 3 parallel datasets as follows:

- **Factual (FA)**: This subset contains triples from the WebNLG and Wikidata datasets.
- **Counterfactual (CFA)**: This subset consists of swapped entities from the factual dataset. These entities are switched based on their class (i.e., a person entity is replaced by another person entity, a date by another date)
- **Fictional (FI)**: This subset consists of made-up entities, obtained via LLM prompting, in place of factual entities.

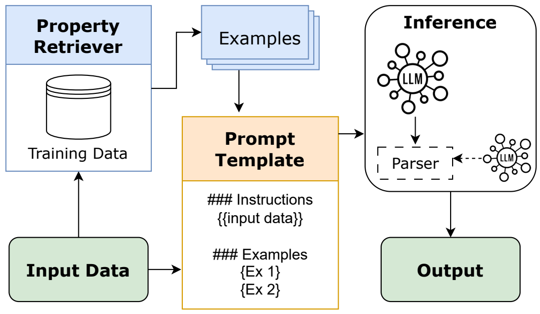


Figure 1: System architecture

Further details and example data for each subtask are available on the shared-task website[4].

3 Method

Our final system consists of a few-shot RAG pipeline that verbalizes the input data. In the following subsections, we describe the details of each component of our RAG pipeline.

3.1 Preprocessing

Our preprocessing step takes an RDF triple as input data and removes unnecessary information from it. For example, the input RDF triple contains the following header:

`<entry category="WikiData human", eid="Id1", shape="unknown", shape_type= "unknown", size="2">`

We realized that the `entry` header does not include any helpful information for verbalization. Hence, we remove it from the input data. We only keep the data between `<modifiedtripleset>` and `</modifiedtripleset>` tags.

3.2 PropertyRetriever

We observed that the verbalization mostly depends on the number of triples and the `Property` fields in the triple. The `Property` field should help in determining the correct verb and verb form. Let's consider the following example.

```
INPUT TRIPLE:

Baked_Alaska | country | France
Baked_Alaska | region | New_York
Baked_Alaska | ingredient | Christmas_pudding

VERBALIZATION:
```

[4] https://gem-benchmark.com/shared_task

```
Christmas pudding is an ingredient
in Baked Alaska, which comes from the
region of New York and the country
of France.
```

For the example above, the model needs to be able to connect all the generated sentences naturally and in a human-like manner. Moreover, it needs to infer the following details:

- *Baked_Alaska* is a food dish based on the property – *ingredient*.
- The property *ingredient* suggests that *Christmas pudding* is an ingredient in *Baked_Alaska*.

In the literature, it is shown that the model can learn to perform such inferences based on few-shot prompting (Lampinen et al., 2022) and retrieval augmented generation (Lewis et al., 2020). Generally, all RAG pipelines use a dense retriever to retrieve relevant samples from the training data. We started by building a dense retriever pipeline using `Haystack` (Pietsch et al., 2019) framework. The dense retriever failed to retrieve examples containing similar properties, especially for the Counterfactual and the Fictional datasets. We realized that this was due to the nature of the dense retriever which is trained to retrieve semantically similar examples. Most of the query input consists of the (*Subject*|*Object*) tokens. Hence, it retrieved examples that are more similar to the query *Subject* and the *Object* tokens.

To solve this issue, we take inspiration from Moryossef et al. (2019) and build a symbolic retriever – `PropertyRetriever`, that retrieves samples from the training based on the most similar properties. The retriever first creates an in-memory index of all properties from the training triples. At query time, it takes the properties of the input triples and returns the best-matching data points. Additionally, these best-matching data points are also selected in a way that the number of properties in the query data and the retrieved data are similar (i.e., shape matching). If no matching properties are found, then the retriever returns the random data points of the same shape. We observed such random sample returns for 130 test points in the WebNLG subtask.

Finally, we compared the verbalizations of 20 input triples using both `PropertyRetriever` and dense retrievers. We find that the samples from the `PropertyRetriever` helped LLMs generate better verbalizations compared to the dense retriever.

3.3 Prompt Engineering

We employ the prompting guidelines provided by the model publishers and Bsharat et al. (2023) for creating our few-shot prompt. We provide our final version of the few-shot prompt in Appendix A.1. Note that the final prompt is a template containing placeholders for the input data and retrieved examples, focusing majorly on task instructions. We use `Banks` (Pippi, 2023) to populate this prompt template with input data and the example data points dynamically at run time.

3.4 Inference

The main goal of our system is to generate suitable verbalization of the input data triples. For the same, we prompt the state-of-the-art LLMs, in a few-shot manner. We use `Mixtral 8x7B` and `Command-R` for all our experiments and compare their performance.

`Mixtral 8x7B`: Mixtral (Jiang et al., 2024) is a decoder-only sparse mixture-of-experts network where the feedforward block picks from a set of 8 distinct groups of parameters. At every layer, for every token, a router network chooses two of these groups (i.e., “experts") to process the token and combine their output additively. This technique increases the number of parameters of a model while controlling cost and latency, as the model only uses a fraction of the total set of parameters per token. Concretely, Mixtral only uses 12.9B parameters per token out of 46.7B total parameters.

`Command R`: Command-R is a 35 billion parameter decoder-only model. It is optimized for conversational interaction and long context tasks. It has been trained with the ability to ground its generations. This means that it can generate responses based on a list of supplied document snippets, and it will include citations in its response indicating the source of the information. This makes it a good candidate for RAG tasks.

Implementation details: We use Ollama[5] to run both `Mixtral`[6] and `Command-R`[7] models locally. We utilize 4-bit quantized versions of these models. We run all our experiments using 2x NVIDIA

[5] https://github.com/ollama/ollama
[6] https://ollama.com/library/mixtral
[7] https://ollama.com/library/command-r

RTX 3090s. The inference hyperparameters are provided in Table 1.

seed	5
temperature	0.5
repeat_penalty	1.2
top_p	0.9
top_k	25

Table 1: Inference Hyperparameters of LLMs

3.5 Postprocessing

We observe that both Mixtral and Command-R cannot follow the formatting instructions perfectly. Zhou et al. (2023) also made similar observations for GPT-4 and PaLM models. We also noticed that it is easier for these models to follow simpler formatting instructions than more complex ones. For example, we initially prompted models to generate verbalization in a JSON format, to which they often made small mistakes such as missing a closing bracket, a semicolon, or a closing quote. We then updated our formatting instructions to just keep the generated verbalization between <verbalization>, </verbalization> tags. After this change, Command-R always generated the verbalization in the correct format and Mixtral's formatting mistakes were reduced significantly.

We parse the model's responses by retrieving the text between <verbalization>, and </verbalization> tags. This way we detect the erroneous responses from both models. We discuss the error patterns in the error analysis section. Finally, we create an ensemble system with both Mixtral and Command-R. For the final output, we use the verbalization generated by the better-performing model (i.e., primary model) if our system can parse the response. Otherwise, we use the verbalization generated by the fallback model (i.e., secondary model) as the final output. We discuss the final choice of primary and secondary models in Section 5.1.

3.6 Evaluation

The system-generated text is assessed with reference-less automatic metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), chrF++ (Popović, 2015), BERTScore (Zhang et al., 2020), and via human evaluation. The criteria for the human evaluation are the following:

- **Grammaticality**: The text is free of grammatical and spelling errors.
- **Fluency**: The text flows well and is easy to read; its parts are connected in a natural way.
- **No-Omissions**: All the information from the input data is present in the text.
- **No-Additions**: Only the information from the input data is present in the text.

4 Shared Task Results

The GEM'24 shared task organizers provide evaluation scores for all participating systems and subtasks using 4 metrics – BLEU, METEOR, chrF++ and BERTScore. These scores are calculated with 1 AMT reference text per data point. Our system achieves the best results on 4 out of 6 subtasks for METEOR and chrF++ metrics. For detailed results and comparison with participating systems, please refer to the overview literature (Mille et al., 2024).

5 Performance Analysis

In this section, we provide the results of our human evaluation study. We conduct this study to finalize our primary and secondary model for the final system. We also discuss observed error patterns during the evaluation.

5.1 Human Evaluation

We conduct a human evaluation study ourselves on a small subset of 40 input triples. These triples are collected from Counterfactual and Fictional datasets. We apply filtering based on our observation that the models generate better verbalization for factual and smaller input triples. Hence, the filtered triples contain more than 3 properties each.

We use the same evaluation criteria mentioned in Section 3.6. We ask our human annotator to rate the model's response on each criterion based on our evaluation guidelines (refer to Appendix A.2). We report the results of this study in Table 2. The results indicate that Mixtral performs better compared to Command-R.

5.2 Error Analysis

We dive deeper into the human evaluation study to figure out exact error patterns. We discuss two of the most commonly observed issues here.

```
INPUT TRIPLE:

What_Ever_Happened_to_Baby_Jane? | publisher | Gruppo_Mondadori
What_Ever_Happened_to_Baby_Jane? | followedBy | I_Am_a_Cat
What_Ever_Happened_to_Baby_Jane? | author | Horst_Köhler
What_Ever_Happened_to_Baby_Jane? | releaseDate | 1726-01-01

GENERATED VERBALIZATION:

The publisher of What Ever Happened to Baby Jane is Gruppo Mondadori. Its author is Horst
Köhler, and it was released on January 1, 1726. Following What Ever Happened to Baby Jane
is I Am a Cat.
```

Figure 2: Example of imperfect verbalization

Criteria	Mixtral	Command-R	Max Score
Fluency	110	105	120
Grammaticality	113	104	120
No Omissions	39	35	40
No Additions	39	38	40

Table 2: Human evaluation scores for the Mixtral and Command-R models.

5.2.1 Fluency issues

We speculate that the fluency issues majorly arise due to the unknown entity type. We provide an example of such an instance in Figure 2. In this case, it is evident that the entity name "What_Ever_Happened_to_Baby_Jane?" or properties (publisher, followedBy, author, releaseDate) may not help in identifying the entity type. Here, the entity may be a movie, book, or literary article. While humans may be able to infer the entity type, we observed cases where LLMs fail to infer entity type or gender from the properties.

5.2.2 Formatting issues

The other major error we observed was that both Mixtral and Command-R can add extra tokens at the beginning of their responses. The most commonly observed beginning tokens for Mixtral are: "It is mentioned that" and for Command-R: "Modified tripleset:". Further, we observe that Mixtral fails to follow the formatting instructions for almost 800 instances out of 1800 total instances. For these 800 instances, we could not extract the verbalization based on our postprocessing steps.

Based on the human evaluation and error analysis results, we choose Mixtral as our primary model and Command-R as the secondary model.

6 Conclusion

In this paper, we describe our solution for the data-to-text generation shared task. We propose a symbolic retriever method – PropertyRetriever, to retrieve better examples for Data-to-text generation problems. We further explore the capabilities of two state-of-the-art LLMs, Mixtral and Command-R. Combining the insights from our human evaluation study and error analysis, we propose an ensemble system as our final solution.

In the future, we would like to explore multi-turn correction and planning approaches. We believe such approaches may allow the model to self-correct its formatting errors and generate verbalizations with better fluency.

Limitations

Our findings require further experimentation on more datasets since we only test our approach on the GEM'24 shared task datasets. We also did not optimize the prompt for each model separately. Optimizing prompts individually for each model can lead to better results. Further, we also use the quantized version of the LLMs which may have affected the accuracy. Our comparison of the 20 samples for deciding between the dense retriever and PropertyRetriever can be further improved by doing a more systematic study. Lastly, our human evaluation study was conducted on a very small subset. For more reliable results, we suggest conducting the human evaluation study on a larger subset.

Ethical Considerations

The human evaluation study presented in this work is carried out by a student assistant at the university. They were paid fairly as per the university payment standards. We also advise evaluating our methods on larger validation sets before using them for other domains and datasets.

Acknowledgments

This research was funded by DFG grant 389792660 as part of TRR 248 – CPEC.[8] We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper. We also thank Sneha Chetani for helping us evaluate the model outputs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 39–54, Prague, Czech Republic. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2023. Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4. *arXiv preprint arXiv:2312.16171*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. In *ACM Comput. Surv.*, volume 55, New York, NY, USA. Association for Computing Machinery.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. 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, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. 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 2267–2277, Minneapolis, Minnesota. 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.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Malte Pietsch, Timo Möller, Bogdan Kostic, Julian Risch, Massimiliano Pippi, Mayank Jobanputra, Sara Zanzottera, Silvano Cerza, Vladimir Blagojevic, Thomas Stadelmann, Tanay Soni, and Sebastian Lee. 2019. Haystack: the end-to-end NLP framework for pragmatic builders.

[8] https://perspicuous-computing.science

Massimiliano Pippi. 2023. Banks: the linguist professor who will help you generate meaningful prompts.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471, Tilburg University, The Netherlands. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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*.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

A.1 Final Prompt

```
###Instruction###: just verbalize the
following data without beginning prompt
in a natural, human-like manner.

###Data### : {{ data }}

###Criteria###: Follow these criteria
carefully:
1. Keep the generated sentences in a flow
and the generated text should sound
human-like.
2. Copy the entities correctly from the data.
3. Replace '\_' with a space.
4. Use the punctuation marks correctly
without any extra spaces.

You may look at the following examples for
the writing style, but only for style.
Do not copy anything from the following
examples, otherwise you will be penalized.

###Examples###:
Ex-1: {{ ex_1 }}
Ex-2: {{ ex_2 }}
Ex-3: {{ ex_3 }}

###Important Notes###:
1. The verbalization output MUST only
contain the verbalization of ###Data###
in a natural, human-like manner.
2. Ensure that generated ###Data###
verbalization MUST be between
<verbalization> and </verbalization> tags.
```

A.2 Human Annotation guidelines

In this task, the model is given data triplets, where each triple is made of *Subject* | *Property* | *Object* and is asked to verbalize this data in a natural, human-like manner.

We need your help to evaluate the model responses based on the following criteria:
Grammaticality: The text is free of grammatical and spelling errors.
Fluency: The text flows well and is easy to read; its parts are connected in a natural way.
No-Omissions: ALL the information in the table is present in the text.
No-Additions: ONLY information from the table is present in the text.

We would like you to focus the most on Fluency and Grammaticality. No-Omissions and No-Additions are binary criteria.
Grammaticality (1-3 scale):

- 1 (Low): The response contains severe grammatical errors that significantly hinder understanding. This may include missing words, subject-verb disagreement, incorrect verb tenses, or nonsensical sentence structure.

- 2 (Medium): The response may contain some grammatical errors, but they are not so severe as to completely obscure the meaning. These errors might include misuse of articles ("a," "an," "the") or prepositions, or minor subject-verb agreement issues.

- 3 (High): The response is free of grammatical errors and adheres to the rules of English grammar.

Fluency (1-3 scale):

- 1 (Low): The response is difficult to read due to awkward phrasing, choppy sentence structure, or lack of variety. It may sound unnatural or unclear.

- 2 (Medium): The response reads mostly smoothly, but there may be occasional awkward phrasing or clunky sentences.

- 3 (High): The response reads effortlessly and sounds natural. The sentences are well-constructed and varied, and the overall flow of ideas is clear and logical.

No-Omissions - Please choose 0 to indicate Missing Information or 1 to indicate No Missing Information.
No-Additions - Please choose 0 to indicate Extra Information/Hallucinated Information or 1 to indicate No Extra Information.

OSU CompLing at the GEM'24 Data-to-Text task

Alyssa Allen and **Ash Lewis** and **Yi-Chien Lin** and **Tomiris Kaumenova** and **Mike White**
{allen.2334, lewis.2799, lin.4434, kaumenova.1, white.1240} @osu.edu

Abstract

This paper details experiments conducted for completing the GEM 2024 Data-to-Text task for a WebNLG dataset (Gardent et al., 2017). We show that model performance varies greatly across English, Spanish, Chinese, and Russian. Data filtering was done with automatic model judgments via error detection, which performs differently per language. We report English and Spanish dev set results for a data filtering and knowledge distillation approach to generating natural language outputs for sets of triples across a variety of domains. Specifically, we compare three generation conditions: 1) few-shot prompting with ChatGPT (GPT4), 2) fine-tuning Llama2 on the unfiltered dataset, and 3) fine-tuning Llama2 on a filtered version of the dataset. Russian and Chinese efforts did not result in submissions due to inconsistent or incoherent translations being produced in either the data synthesis or final generation stages. We provide details on these shortcomings but largely focus on Spanish and English efforts that align with our task submissions. We ultimately submitted outputs in English and Spanish that were generated using a version of Llama2 fine-tuned on a filtered dataset.

1 Introduction

In the WebNLG 2020 Challenge, the OSU system (Li et al., 2020) and others achieved nearly flawless English performance by carefully fine-tuning a pretrained model on the training set, though performance in Russian was remarkably poor by comparison. Since then, large language models (LLMs) like OpenAI's ChatGPT, Anthropic's Claude, and Google's Gemini have exhibited increasingly remarkable performance on a wide variety of tasks in multiple languages using in-context learning (i.e., few-shot generation), potentially obviating the need for human annotated training data. In this work, we first examine ChatGPT's few-shot performance on this data-to-text task — converting WebNLG-style logical triples to text — in a variety of languages and find several limitations in cases with lower-resource languages. Coupling these limitations with the high computational and financial costs, as well as lack of consistent behavior across time, makes ChatGPT difficult to rely upon for such tasks across languages. In this work, we leverage the robust capabilities of ChatGPT but attempt to offset the aforementioned downsides via knowledge distillation with smaller, more economical open source models.

Following the example of Lewis and White (2023), we leverage few-shot prompting to first generate training data using the ChatGPT API, then to detect errors in the generated data and filter them out. The filtered and unfiltered datasets are then used to fine-tune a Llama-2 (Touvron et al., 2023) model for the data-to-text task. This strategy was employed for English, Spanish, Russian, and Chinese data but we ultimately found that it was not effective for the latter two.

Specifically, Chinese fine-tuning Llama2 (pre-trained on Chinese) on the initial filtered and unfiltered datasets yielded inconsistent outputs. This paper therefore describe alternative data filtering approaches and initial dev set results.

On the other hand, Russian experimentation was the least fruitful as there were numerous errors in the data synthesis and filtering stages. This paper describes some of the obstacles faces for this language and details our efforts to resolve these issues.

This work shows that filtering synthesized training data (for Spanish and English conditions) is a promising strategy for generating fluent, informative, and concise outputs. Initial dev set results indicate that improvements to our filtration methods would ultimately maximize performance. By employing a knowledge distillation approach, we capitalize on ChatGPT's expertise while lowering overall cost of generation by using an open-source

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 100–111
September 23 –27, 2024. ©2024 Association for Computational Linguistics

model like Llama2.

Our work also describes shortcomings of LLMs in completing this task for Chinese and Russian. By describing various data filtering efforts, we show potential paths forward in generating fluent and grammatically correct outputs across languages.

2 Related Work

The strategy of knowledge distillation in which we use ChatGPT to generate synthetic data, apply automatic data filtering methods, and use the filtered data to train a downstream model is most closely related to the work of Lewis and White (2023) in which they use the same general approach to create a virtual tour guide for a museum. Kim et al. (2023) use GPT-3 to construct a conversational dataset for distilling a smaller T5 conversational model in a similar fashion, focusing on social conversation filtered for safety. Lewis and White's (2023) setting involves retrieval augmented generation (RAG) and both of these works involve chat functions, while our work here applies knowledge distillation and filtering in a data-to-text task.

Schneider et al. (2024) explore the viability of using LLMs, both with few-shot prompting alone and fine-tuning, for the task of semantic parsing in a conversational setting and find that the recent models are able to perform reasonably well on the task, though they investigate English only. Our work also explores performance on other languages.

Madaan et al. (2023) use a self-refine approach to refine synthetic data, which, while not immediately relevant to this work, aligns with our efforts to refine synthetic data to improve the overall training set. Other works that have utilized knowledge distillation for seq2seq models include Tang et al. (2019) and Chen et al. (2020), who both explore distilling BERT's bidirectional encoder knowledge into a seq2seq model for generation. Our method can also be viewed as the first step in an approach combining knowledge distillation with self-training, not unlike Heidari et al. (2021), who implement self-training using an acceptability classifier (Batra et al., 2021) and ultimately distill fine-tuned BART (Lewis et al., 2019) seq2seq models. More recent approaches also explores using LLMs in self-training with various refinements to how acceptability classifiers are incorporated (Gulcehre et al., 2023; Yuan et al., 2024).

3 Methodology

We used the WebNLG dataset (Gardent et al., 2017) to synthesize a training set, including factual, counterfactual, and fictional examples. We then experimented with few-shot prompting, data filtering, and knowledge distillation via fine-tuning.

Target languages were English, Spanish, Chinese, and Russian. This section provides an overview of methods that apply to all target languages. Section 4 details language-specific efforts that vary depending on problem cases. Section 4 for Spanish and English also provides automatic evaluations on dev set results and initial insights based on those findings.

3.1 Dataset Conditions

This describes how each training data for each task condition is generated. Factual examples include sets of triples given in the WebNLG dataset (Gardent et al., 2017) across a variety of domains. For fictional and counterfactual data synthesis, we followed processes and definition outlined by the GEM task description. Our tactics differ from Axelsson and Skantze's (2023) knowledge-graph approach in that we chose to leverage the same ideas of entity relationships as is present in the aforementioned knowledge-graph approach, but manually assigned relationship (which we'll call predicate types) in order to more quickly generate our training data for this task.[1]

Fictional and Counterfactual examples were created by first manually assigning a type for each unique predicate (or second element) in a factual data set triple. After each unique triple across domains in the dataset were identified, a sample triple was selected randomly for that predicate. We then used our best judgment to create general categories that reflect the type of element needed to accompany the predicate. For example, if the predicate is `birthDate`, the author doing the annotations would see a triple such as `Alan_Bean | birthDate | 1932-03-15`. The annotator then assigns predicate `birthDate` with type `Person, Location`.

A predicate type consists of a Subject and Object label, indicating the relationship between the first and third element within a triple with its predicate. This predicate type indicates that the first element

[1]Had the organizers made synthetic input triples available for development, or code to generate such triples, it would have made it possible to spend more of the limited development time on the text generation portion of the task.

(i.e., subject) needs to be a person and the third element (i.e., object) should be a location.

After each unique predicate (372 predicates) is assigned a type, a set of fictional elements for each subject and object is made. For each unique subject and object label, we prompt ChatGPT (gpt-3.5-turbo). From the example above, the prompts would be 'Generate 20 realistic sounding People' and 'Generate 20 realistic sounding Locations'. Using the subject and object labels for each generated list, fictional elements can be swapped into the original factual triple sets. See Appendix A for more details.

Counterfactual examples were created by using a similar method, except instead of prompting ChatGPT, the list of potential element replacements were created by keeping track of first elements with the same subject or object label. Any first element of a given type could be swapped with another first element of that type. The same was done for elements in the third position in the triple.

Due to time constraints, type assignments did not change across domains per predicate. Therefore, some counterfactual and fictional swaps yielded less intuitive triple sets. For example, `creator` is a predicate in two domains: 'Food' and 'ComicsCharacter'. If creator is given the predicate type `Food, Person` in the labeling process, the swap could result in one triple relating to food and the rest if the triple set referring to a show. This disconnect makes the model less likely to generate concise and fluent outputs. Ultimately, these examples were not noticeably evaluated differently by the model than more logical triple sets. These annotations were also used to gain a rough sense of auto-evaluation accuracy. Further refinement of our swapping processes could help elevate the quality of our training data.

3.2 Data Synthesis and Knowledge Distillation

Using the training data, we prompt ChatGPT (3.5-turbo or 4 depending on target language) to generate natural language sentences per triple set. Instructions were given in the target language (see Appendix B). Dataset triples were all in English. Table 1 shows training set sizes for each language. Unfiltered, synthesized training sets for English, Chinese, and Russian include 1500 examples. Spanish training set size was 8,643 examples (or 20% of all possible examples provided in the WebNLG dataset). The Spanish data is more robust than the other languages because we found initial success generating coherent outputs, but wanted to test how more data could improve model performance for Spanish. Early experimentation with Russian and Chinese was less successful and therefore more time needed to be spent on process versus data quantity. We acknowledge that OpenAI does not provide a way to produce outputs in a reproducible manner. To increase transparency in our process that uses ChatGPT outputs as training data, we made the synthetic data generated by ChatGPT publicly available on GitHub.[2]

Filtering should exclude only examples with poor quality. However, the automatic filtering method described in Section 3.3 is noisy, cutting out a lot of good examples along with bad ones. Thus, one would expect that the larger the size of the training set, the less it matters to mistakenly cut out some good items, as there remain plenty of others, leaving more room to see a benefit of using cleaner data with relatively fewer bad items remaining. Based on the data set sizes in Table 1, we expect filtering to work best on Spanish since it is nearly 6x larger than Chinese and English data sets.

We fine-tuned Llama2 on this data set to provide us with a baseline ahead of filtering (see Section 3.3). As discussed in Section 4, each language leveraged various pretrained versions of Llama2.

3.3 Data Filtering via Error Detection

Following data synthesis, we experimented with filtering the training data discussed in Section 3.2. We used ChatGPT (GPT 4) as an error detector where we asked the model to determine if a generated text faithfully and fluently corresponded to a given triple set (see Appendix C for full prompts).

To evaluate the error detection capabilities of ChatGPT, the first author manually annotated a small set of 99 examples across categories (factual, counterfactual, and fictional) for English and Spanish. 8 errors were found in Spanish and 9 errors were found in English.

Out of 99 Spanish examples, ChatGPT's (GPT4) performance as an error detector yielded 50% recall and 36% precision (see Table 2). Out of 99 English examples, error detection yielded 44% recall and 31% precision. These numbers exceed the actual error rate of about 10% in both languages which means the error detector performance is far from ideal but well above chance.

[2] https://github.com/nicalin/2024-GEM-data

Language	Unfilt. Size	Filt. Size	Filt. FA	Filt. CFA	Filt. FI
English	1,500	1,315	500	378	437
Chinese	1,500	1,172	463	324	385
Spanish	8,643	7,607	2,735	2,286	2,586

Table 1: Size of unfiltered and filtered training data sets. For filtered data, this table also shows how many examples were included across languages and per category — factual (FA), counterfactual (CFA), fictional (FI).

	Agreement (%)				Error Detection		
Language	Overall	FA	CFA	FI	Recall	Precision	F-Score
English	87	90	80	90	0.44	0.31	0.36
Spanish	88	91	76	97	0.50	0.36	0.42

Table 2: Agreement and error detection results for training data sets across languages and triple set categories — factual (FA), counterfactual (CFA), fictional (FI) — for model-based judgments.

As seen in Table 2, counterfactual cases had the lowest accuracy in both English and Spanish. Qualitatively, the model labeled correct counterfactual cases as incorrect when it wanted to correct the facts or if there were conflicting dates in the triple set. Consider the example in Figure 1.

In the above example, the generated text is faithful to the information in the triples. The evaluator is unable to ignore real-world knowledge. This output is deemed as incorrect because of factual contradictions.

Performance could likely be improved with more nuanced evaluation prompts or a refined process for creating the counterfactual triples. Time was a constraint in making these adjustments.

Fictional cases had higher agreement than expected due to fewer errors being in the fictional category compared to factual or counterfactual. Therefore, the agreement between model and author was primarily for correct cases instead of on errors.

Cases that passed the filter are added to a validated dataset. This version of the training data is then used to fine-tune Llama2. Russian experimentation did not successfully complete the filtering stage (discussed in Section 4.4). Outputs from Llama2 (fine-tuned on the filtered data) for English and Spanish were ultimately submitted.

4 Experiments

This section details language-specific efforts and offers further insights as to how model performance differed per language.

4.1 English

Experimentation with English on data synthesis and filtering followed the methods described in Section 3. This section describes dev set results for English. The dev set consists of 1,998 examples (666 per category). Our English submission for this task was generated using Llama2 fine-tuned on the filtered dataset.

4.1.1 Automatic Evaluation

Since our dev set did not have gold standard outputs, we could not use reference-based methods for automatic evaluation. We instead employed the same error detection method as we used for filtering (Section 3.3) in order to compare experimental methods within a target language. Few-shot prompting with Llama2 yielded inconsistent outputs. For example, common output errors were regurgitating the triple set and printing '\n' repeatedly. These errors were not present in the fine-tuning conditions. Table 3 shows the percentage of 'good' outputs per category. As shown in Table 3, dev set results are nearly equal for unfiltered and filtered Llama2 fine-tuning conditions with 77% and 75% (respectively) of the outputs being judged as faithful to their corresponding triple sets.

4.1.2 Analysis

To quickly check efforts during the task, we performed an automatic evaluation on a small dev set of about 120 items that led us to believe a filtering approach would yield higher quality results. Results shown in Table 3 were conducted after the task deadline. Factual cases across conditions outperformed counterfactual and fictional categories. Counterfactual cases across conditions showed the

TRIPLES: `France | country | Jinnah_International_Airport`
GENERATED TEXT: Francia es el país donde se encuentra ubicado el Aeropuerto Internacional Jinnah.
ENG TRANSLATION: France is the country where Jinnah International Airport is located.
CHATGPT JUDGMENT: Bad. The triple suggests that Jinnah International Airport is in France, which is incorrect. Jinnah International Airport is in Pakistan, not in France.
ANNOTATOR JUDGMENT: Good.

Figure 1: Example of a generated output in Spanish that is considered by the authors to be a 'good' example, but the evaluator deemed it 'bad'. As seen in the CHATGPT JUDGMENT, the evaluator labels the output as incorrect due to the counterfactual nature of the triple.

	'good' Judgments (%)			
Condition	**Overall**	**FA**	**CFA**	**FI**
Eng_Unfiltered_LL	77	91	65	74
Eng_Filtered_LL	75	88	65	72
Eng_FewShot_GPT	84	97	72	83

Table 3: Automatic evaluation English dev set results for filtered (Llama2), unfiltered (Llama2), and few shot (GPT4) conditions. Percentages represent how many generated outputs were judged as faithful to its triple set input. Results are shown overall and across categories Factual (FA), Counterfactual (CFA), and Fictional (FI).

	'good' Judgments (%)			
Condition	**Overall**	**FA**	**CFA**	**FI**
Esp_Unfiltered_LL	79	92	71	75
Esp_Filtered_LL	81	92	72	77
Esp_FewShot_GPT	89	98	82	88

Table 4: Automatic evaluation for Spanish dev set for filtered (Llama2), unfiltered (Llama2), and few shot (GPT4) conditions. Percentages represent how many generated outputs were judged as faithful to its triple set input. Results are shown overall and across categories Factual (FA), Counterfactual (CFA), and Fictional (FI).

worst performance. Both Llama2 conditions performed worse than ChatGPT.

Further experimentation with filtering is needed to improve over an unfiltered fine-tuning approach. Overall, filtering the data slightly decreased 'good' judgments of English outputs, from 77% to 75%. As mentioned in Section 3.2, the smaller the training dataset is, the more a noisy filtering method will likely hinder performance results. Therefore, the filtering method was not as effective on the smaller English dataset as it was on the larger Spanish dataset (see Table 1).

We expect that a more carefully orchestrated filtration methodology could yield better results and further work explore better ways to utilize LLMs for this task via prompting or perhaps by other fine-tuning strategies. Based on results in Section 4.2.1, where the filtering condition improved upon the unfiltered condition, we expect that increasing the training data size for English would improve results for the filtered condition.

Furthermore, we find that using ChatGPT as an evaluator is helpful for obtaining rapid judgments but is not fully reliable. Table 3.3 shows that agreement with an annotator (the first author) is high but imperfect, particularly in the counterfactual cases. Because we used ChatGPT in a similar way for our filtration methodology, it follows that it too is imperfect. Figure 2 shows an example of disagreement between annotator and model judgments for a counterfactual case.

Because we found lower agreement with author judgments for the CFA items (see Table 2), we expect the error rate to be somewhat inflated in these automatic evaluation results shown in Table 4. That said, we still expect the CFA items to be the most difficult, and cursory inspection reveals plenty of real errors in the text, as seen in Figure 2. Further experimentation with explicitly prompting the system to ignore real-world facts could lead to improved CFA case results.

4.2 Spanish

Experimentation with Spanish followed the methods described in Section 3. We used a version of Llama2 pretrained on Spanish[3] for all Spanish ex-

[3] https://huggingface.co/clibrain/Llama-2-7b-ft-instruct-es

TRIPLE: `Chicago | isPartOf | Linn_County,_Oregon`
TEXT: Chicago is part of Linn County, Oregon.
CHATGPT JUDGMENT: Bad. There is a geographical inaccuracy as Chicago is not part of Linn County, Oregon; it is in fact, a city in Illinois. Therefore, the information provided in the text contradicts established facts and cannot be corrected without changing the original triples.
ANNOTATOR JUDGMENT: Good.

Figure 2: Example of a generated output in English that is considered by the authors to be a 'good' example, but the evaluator deemed it 'bad'. As seen in the CHATGPT JUDGMENT, the evaluator labels the output as incorrect due to the counterfactual nature of the triple.

periments. Our Spanish submission for this task was generated using Llama2 fine-tuned on the filtered dataset.

This section describes the Spanish dev set results. As described in Section 3.1, factual inputs are compiled from the WebNLG dev set (Gardent et al., 2017) and we synthesized examples for fictional and counterfactual categories. Our dev set consists of 1,998 examples (666 per category).

4.2.1 Automatic Evaluation

Error detection (described in Section 3.3) was used for data filtering. Few-shot prompting with Llama2 yielded inconsistent outputs. As seen in Table 4, fine-tuning with our full training set yielded 79% 'good' judgments during the automatic evaluation. Fine-tuning on a filtered dataset yielded 81% 'good' judgments. Using ChatGPT (GPT4) to generate outputs yielded 89% 'good' judgments.

Unlike the English case, this method resulted in some improvement for counterfactual and fictional conditions, thereby increasing the overall accuracy in Spanish.

4.2.2 Analysis

Similar to the process mentioned in Section 4.1.2, this evaluation was conducted on a substantial dev set after the deadline for this task. Within the scope of the task, we conducted an initial automatic evaluation on a small dev set of about 90 examples, where results were comparable for the two fine-tuning conditions.

Factual cases yielded the highest percentage of 'good' cases across conditions. Counterfactual cases were also the worst performing across conditions. This trend is consistent with trends found in English results, see Table 4.

Results for the GPT condition shown in Table 3 and Table 4 are surprising given English's presumed greater prevalence in ChatGPT's training data. The reason for Spanish results outperforming English results is unknown.

Of note, as mentioned in Section 3.2, the Spanish training dataset was significantly larger than the training set used for any other language (see Table 1). This increase in training data could have led to overall improvement in filtered (79%) vs. non-filtered (81%) conditions for Spanish, but not English. To further improve filtered condition performance, further refinement of filtration methodology is needed. More training data could also improve results for both filtered and unfiltered model conditions.

4.3 Chinese

For Chinese, we used the same data synthesis methods as described in Section 3. We ultimately attempted to fine-tune a version of Llama2 trained on Chinese (Zefeng Du, 2023) with filtered Chinese data, but were unsuccessful in generating consistent, high-quality results. One recurring issue with dev set results was that Llama2 would provide additional narration to the outputs such as: '好的，让我来给您介绍一下... (English translation: Okay, let me introduce you to...)'.

Due to the time constraints, we were not able to investigate the cause of this issue or otherwise improve our fine-tuned model. While we did not submit Chinese results to this task, this section details experimentation with data filtering for Chinese.

4.3.1 Chinese Data Synthesis and Filtering

As mentioned in Section 3, we used ChatGPT (GPT 4) for data synthesis and filtering via few-shot prompting. In addition to generating one Chinese output per triple set (One-to-One), we experimented with generating five Chinese outputs per triple set (One-to-Many), with the aim of increasing the size of the filtered dataset (see Section 3.3 for error detection and filtering methods).

We also experimented with different data filtering methods using GPT 4 for One-to-One and One-to-Many datasets. For One-to-One, we experimented with two data filtering methods: (1) error detection and (2) reconstruction. For One-to-Many, we experimented with likelihood rankings.

Filtering via Error Detection

Filtering via error detection leverages methods outlines in Section 3.3. We provided ChatGPT with a triple set and a Chinese output. We then prompted ChatGPT to judge outputs as 'good' or 'bad' (See Appendix C for full prompt). In addition to asking ChatGPT to provide judgments, we also experimented with two different settings: (1) asking ChatGPT to only provide the judgment (i.e., 'good' or 'bad') and (2) asking ChatGPT to provide a correction for any output labeled 'bad'.

Filtering via Reconstruction

For filtering via reconstruction, we employed ChatGPT as a reverse model to reconstruct the English triple set from a given Chinese synthesized output. In principle, if the reconstructed English triple sets are overly different (see Section 4.3.2) from the corresponding original triple sets, the synthesized outputs are likely to have poor quality and should be discarded.

Filtering via Likelihoods

For the One-to-Many dataset, we used the ChatGPT `logprobs` parameter to assign a pair likelihood scores to each output per triple set. Specifically, we treated this process as a binary classification task. Each candidate output per triple set received two log probability scores (likelihood that output is labeled 'good' and likelihood that output is labeled 'bad'). The output with the highest 'good' score compared to other candidate outputs for a given triple set was kept as the output for that triple set. An output was selected at random if multiple candidates received the same score.

4.3.2 Automatic Evaluation

Automatic evaluation for each filtering method discussed in Section 4.3.1 follows error detection methods as described in Section 3.3. In order to justify automatic evaluation results, we compare model judgments with author judgments. For each data filtering method, we manually annotated 30 examples to compare filtering method performance. The sets of annotated examples differ per filtering method. Therefore, the comparisons in Table 5 are not directly comparable.

We excluded the odd cases in the 30 examples, resulting in 29, 30, 29, and 30 annotated examples respectively for model judgment with correction (Judge/Corr), model judgment without correction (Only_Judge), reconstruction, and likelihood. Table 5 shows precision, recall, and agreement resulting from author judgment compared to ChatGPT's judgment.

Judge/Corr and Only_Judge model outputs are synonymous with filtering method judgments. For reconstruction, we did not reach a stage of automatic filtering based on similarity of the reconstructed triple set to the original triple set. Therefore, the annotator manually judged each reconstructed triple set. Reconstructed triple sets deemed as faithful to the original triple set are expected to yield faithful natural language outputs as well. These judgments on the reconstructed triples (i.e., acting as pseudo–model judgments) are compared to the original author annotations.

For filtering via likelihood, an output is labeled 'bad' if it has a higher likelihood for 'bad' instead of 'good'. Similarly, the output is labeled 'good' if it has a higher likelihood for 'good' instead of 'bad'. These judgments serve as the model judgments to compare with the author annotations.

Table 5 shows results of model agreement with the manual annotations. As shown in Table 5, filtering with reconstruction yielded the best precision and recall, albeit with manual similarity judgment of triple sets.

4.3.3 Analysis

In the Judge/Corr condition, outputs which were incorrectly judged as 'bad' were either counterfactual or fictional. Cases judged incorrectly as 'good' were most often due to missing information in the triple sets, hallucination, or incorrectly representing the relation.

In the Only_Judge condition, the outputs incorrectly judged as 'bad' by ChatGPT were counterfactual and were often due to the model fact-checking the output. Outputs with incorrect 'good' judgments were mostly due to incorrect or disfluent translations of the relations, which were not detected by ChatGPT.

In the reconstruction condition, outputs which were incorrectly judged as 'bad' were across all three categories, mainly due to the reconstructed triple set missing information. The reconstruction condition also incorrectly judged some outputs as 'good', mainly because the reconstructed triple sets were not able to show disfluency and incorrect translation in the synthesized sentences.

In order to automate the reconstruction filtering

Filtering Method	Precision	Recall	Agreement (%)
Judge/Corr	0.33	0.50	79
Only_Judge	0.50	0.40	83
Reconstruction	0.63	0.56	76
Likelihood	0.33	0.25	83

Table 5: Precision, recall, and agreement of the different data filtering methods for the Chinese data. Precision and recall are for error detection. Judge/Corr, Only_Judge, and Reconstruction methods are One-to-One. Likelihood method is One-to-Many.

process, we would need a process for comparing the reconstructed triple set to the original triple set. Additionally, an added complication of Chinese is that some Chinese terms can have multiple English translations. Therefore, translating from English to Chinese and then back to English leaves increasing room for error. Using LLMs to automating the process of comparison could be a potential direction to explore in future work.

Lastly, for the likelihood condition, the final candidates incorrectly detected as errors are either counterfactual or fictional. The main error found in this case was incorrect translations and missing information.

Due to time constraints, data filtering via error detection (without correction) was the filtering method ultimately used to create the filtered Chinese training data for fine-tuning. However, given more time to refine processes, filtering via reconstruction could potentially improve model performance.

4.4 Russian

We employed few-shot prompting to synthesize Russian outputs for the training data triple sets. This method involved providing GPT-4 with 5 examples (1 factual, 2 counterfactual and 2 fictional) in Russian. The model generated sentences often exhibited unnatural phrasing, incorrect case endings, as well as inconsistencies and inaccuracies in translating proper names (see Appendix D).

Grammatical issues persisted after attempting to repair errors via automatic judgment and error correction. We provided ChatGPT with a few-shot prompt where the examples included faulty outputs, suggested corrections, and then the corrected version of the outputs. We also prompted ChatGPT to make judgments without repairing the errors. Instructions were given in Russian. The model failed to consistently detect and correct errors in the synthesized data. Low model performance indicates the need for more sophisticated techniques and training datasets to improve the quality of Russian text generation in future work.

4.4.1 Analysis

Our primary approach for synthesizing Russian data was few-shot prompting which did not yield satisfactory results. The attempt to use model-generated judgments to filter and correct outputs did not sufficiently mitigate the issue of fluency. In future work, advanced translation models could assist in synthesizing Russian data by translating English outputs. Additionally, further experimentation with fine-tuning approaches using a small manually constructed set of Russian examples could improve the quality of generated sentences.

5 Preliminary English Results

The task organizers provided preliminary results for the English data set (Mille et al., 2024). Results are for factual, counterfactual, and fictional cases for the WebNLG test set (i.e., the 'seen' subtask) and the WikiData test set (i.e., the 'unseen' subtask).

As expected, our team's best performing category was WebNLG FA with a BLEU score of 30.03. WebNLG FI performed worse than WebNLG CFA with BLEU scores of 21.44 and 24.45, respectively.

The seen results for FA and FI results were higher than the unseen ones in the same categories with BLEU scores of 24.97 for unseen FA and 16.9 for unseen FI cases.

Given aforementioned challenges with our CFA training data in terms of potentially nonsensical triple sets, it is not surprising that CFA unseen BLEU score (27.06) outperformed the seen BLEU score (24.45).

Fictional cases were our lowest performing for both subtasks. This may be due to the system's desire to adhere to real-world facts. As mentioned in Section 4.1.2, more explicit prompting could have led to improved performance.

Compared to other systems that completed this data-to-text task, our system ranked in the middle across categories. A difference that potentially led other systems to outperform our system is that we opted out of using existing supervised training data. Instead, we chose to use a limited amount of few-shot synthetic data generated by ChatGPT. Because our efforts expanded across English, Spanish, Chinese, and Russian, we avoided tactics that may give English an advantage over the other languages of interest (for which the same supervised training data was not available). We could have also experimented with generating more few-shot synthetic data and using self-training methods to improve our system's performance but did not due to expense and lack of time.

While we did not have access to Spanish results at the time of publishing, we expect overall improved performance due to increased training data set size compared to the English training set.

6 Discussion and Conclusion

For all languages, newer state of the art models such as Llama3 (AI@Meta, 2024) could have improved output performance — specifically in the cases of Russian and Chinese where incorrect translations were a large portion of errors qualitatively found.

We also chose to not mention in the evaluation prompts how to handle non-factual triple sets and corresponding output being tested. The evaluations for counterfactual and fictional cases could potentially have been improved if the prompt included explicit instructions to ignore any non-factual information and focus on the representation of the triple sets in the output.

A data synthesis and knowledge distillation approach yielded promising results in English and Spanish. ChatGPT was successful in synthesizing training data, but was less-than-ideal in acting as an evaluator — particularly for Chinese. ChatGPT was less successful in generating usable synthetic data for Russian.

Further work could focus on refining our approach to data filtering and experiment with self-training, which could potentially yield results comparable to or even exceeding few-shot prompting with ChatGPT using a cheaper and more reliable open source model. Based on the experiments presented in this paper, we see that LLMs perform better on data-to-text tasks for higher resource languages (English and Spanish) and struggle with others (Chinese and Russian).

References

AI@Meta. 2024. Llama 3 model card.

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. *Preprint*, arXiv:2307.07312.

Soumya Batra, Shashank Jain, Peyman Heidari, Ankit Arun, Catharine Youngs, Xintong Li, Pinar Donmez, Shawn Mei, Shiunzu Kuo, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Building adaptive acceptability classifiers for neural NLG. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 682–697, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188. Association for Computational Linguistics.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling. *Preprint*, arXiv:2308.08998.

Peyman Heidari, Arash Einolghozati, Shashank Jain, Soumya Batra, Lee Callender, Ankit Arun, Shawn Mei, Sonal Gupta, Pinar Donmez, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Getting to production with few-shot natural language generation models. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 66–76, Singapore and Online. Association for Computational Linguistics.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. Soda: Million-scale dialogue distillation with social commonsense contextualization. *Preprint*, arXiv:2212.10465.

Ashley Lewis and Michael White. 2023. Mitigating harms of LLMs via knowledge distillation for a virtual museum tour guide. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*, pages 31–45, Prague, Czech Republic. Association for Computational Linguistics.

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*.

Xintong Li, Aleksandre Maskharashvili, Symon Jory Stevens-Guille, and Michael White. 2020. Leveraging large pretrained models for WebNLG 2020. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 117–124, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Phillip Schneider, Manuel Klettner, Kristiina Jokinen, Elena Simperl, and Florian Matthes. 2024. Evaluating large language models in semantic parsing for conversational question answering over knowledge graphs. *Preprint*, arXiv:2401.01711.

Raphael Tang, Yao Lu, and Jimmy Lin. 2019. Natural language generation for effective knowledge distillation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 202–208, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Longyue Wang Zefeng Du, Minghao Wu. 2023. Chinese-llama-2. `https://github.com/longyuewangdcu/Chinese-Llama-2`.

7 Appendices

A Fictional Element Prompts

The following prompts showcase the process of creating fictional examples.

Example Predicate types

Examples are of the form - predicate: [Subject label, Object label] foundingDate: [Company, Date] author: [Book, Author] starring: [Film, Person]

Example Prompt

Each label is then substituted into the following prompt 'Generate 20 realistic-sounding...'.

Creating Fictional Examples

After prompting ChatGPT, each subject or object label has a list of options. For each example in the factual dataset, the appropriate labels are found per predicate and an item from the fictional element lists for that label is selected at random. Substitions are then made accordingly. For example:

Factual Triple

1. `Siomay | ingredient | Peanut sauce`
2. `Batagor | dishVariation | Siomay`

Predicate Types

1. ingredient: `Dish, Ingredient`
2. dishVariation: `Dish, Dish`

Fictional Triple

1. `Beef enchiladas with cilantro lime rice | ingredient | Almond flour`
2. `Penne pasta in creamy tomato sauce | dishVariation | Mediterranean Chickpea Salad`

B Generation Prompts

The following prompts were used for each language of interest. Native speakers wrote the English, Chinese, and Russian prompts. Spanish prompt was written by an L2 Spanish speaker.

Additionally, Chinese prompt includes both instructions and few-shot examples. For the Chinese prompt, only instructions are included here. For Russian, since our Russian prompt is a few-shot prompt consisting of 5 translated examples without explicit instructions, we do not list the prompt here.

English

Write a text version of the info in the input. The text should include all of the information in the triples, and only that information. The output should be fluent, grammatically correct, and concise.

Spanish

Analiza los siguientes ejemplos de tríos y textos. El texto debe incluir toda la información de los tríos y solo esa información. Además, el texto debe ser fluido, gramaticalmente correcto y conciso. Escribe el texto para el caso de prueba.

Chinese

请把以下每组三词词组转换成中文文字。请不要省略任何信息或是增加任何不必要的信息。有些三词词组可能不符合事实，但请还是把提供的三词词组转换成中文。

C Error Detection Prompts

The following instructions were used with ChatGPT to evaluate the generated texts. Prompts for English and Spanish evaluation were in English. Results were used to determine which cases pass the error detection filter.

English and Spanish

Examine the following triples and [Spanish] text. The [Spanish] text should include all the information in the triples without adding unnecessary information. The [Spanish] text should be fluent, grammatically correct, and as concise as possible. If the [Spanish] text is all correct, respond with 'Good.' If not, respond with 'Bad.' and explain the error and then rewrite the answer.

Chinese

For Chinese, there are four different prompts each filtering method (i.e., automatic judgment (with correction), automatic judgment (without correction), likelihood, and reconstruction). The instructions for the first three filtering methods are in Chinese while instruction for reconstruction is in English.

Error Detection with Correction

检查以下的内容以及正文。管有些容及正文可能不符合事，正文是应该包含所有内容里的资讯，并且正文不能包含不必要的资讯。正文必须流畅且文法正确。如果判断正文为完全正确，回覆'好'。如果有任何错误，回答'坏'，并且解释错误以及修改正文。

Error Detection without Correction

检查以下的内容以及正文。管有些容及正文可能不符合事，正文是应该包含所有内容里的资讯，并且正文不能包含不必要的资讯。正文必须流畅且文法正确。如果判断正文为完全正确，回覆'好'。如果有任何错误，回答'坏'。

Likelihood

请根据以下的内容检查每个句子。尽管有些内容及句子可能不符合事实，正文是应该包含所有内容里的资讯，并且正文不能包含不必要的资讯。正文必须流畅且文法正确。如果判断正文为完全正确，将句子归类到'好'的类别。如果有任何错误，将句子归类到'坏'的类别。

Reconstruction

Convert the following Chinese text into a set of English triples. The Chinese text might be counterfactual. Please still try to convert the Chinese text into English triples.

D Russian Errors

Russian

1. As shown in the example, the model exhibited unnatural phrasing.

> **GENERATED TEXT:** Эллиот Си был летчиком-испытателем и был выбран NASA в 1962 году. Он родился 23 июля 1927 года и умер 28 февраля 1966 года в Сент-Луисе. Сейчас его статус - умерший.
>
> **ENG TRANSLATION:** Elliot See was a test pilot and was selected by NASA in 1962. He was born on July 23, 1927 and died on February 28, 1966 in St. Louis. Currently his status is dead.
>
> **CORRECTED TEXT:** Эллиот Си был выбран в NASA

летчиком-испытателем в 1962 году. Он родился 23 июля 1927 года и умер 28 февраля 1966 года в Сент-Луисе.

2. Some words, especially proper nouns, were not properly translated.

   GENERATED TEXT: Эллиот Си, родившийся 23 июля 1927 года в Далласе, был гражданином США. Он занимался профессией тестового пилота. Си учился в Университете Техаса в Остине, который состоит в Большой 12 конференции. Эллиот Си скончался в Сент-Луисе.

   ENG TRANSLATION: Elliot See born on July 23, 1927 in Dallas was an American citizen. He worked as a test pilot. See studied at the University of Texas at Austin which is a part of Big 12 conference. Elliot See died in St. Louis.

   CORRECTED TEXT: Американец Эллиот Си, родившийся 23 июля 1927 года в Далласе, был летчиком-испытателем. Си учился в Университете Техаса в Остине, который входит в состав конференции Big 12. Эллиот Си скончался в Сент-Луисе.

3. Case endings were often generated incorrectly.

   GENERATED TEXT: Техническим кампусом института технологий Ачарьи был признан Всеиндийским советом по техническому образованию (AICTE), расположенным в Мумбаи. Институт находится в городе Бангалор, штат Карнатака, который граничит с Аравийским морем на западе. Институт Ачарьи аффилирован с Вишвесварайским технологическим университетом.

   ENG TRANSLATION: The Acharya Institute of Technology was recognized as the 'Technical Campus' by the All India Council for Technical Education in Mumbai. The institute is located in Bangalore, Karnataka, which borders the Arabian Sea to its west. The Acharya Institute is affiliated with the Visvesvaraya Technological University.

   CORRECTED TEXT: Институт технологий Ачарьи получил статус технического кампуса от Всеиндийского совета по техническому образованию (AICTE), расположенного в Мумбаи. Институт находится в городе Бангалор, штат Карнатака, который граничит с Аравийским морем на западе. Институт Ачарьи является филиалом Вишвесварайского технологического университета.

CUET_SSTM at the GEM'24 Summarization Task: Integration of Extractive and Abstractive Method for Long Text Summarization in Swahili Language

Samia Rahman, Momtazul Arefin Labib, Hasan Murad, Udoy Das
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u1904022, u1904111}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd,
u1804109@student.cuet.ac.bd

Abstract

Swahili, spoken by around 200 million people primarily in Tanzania and Kenya, has been the focus of our research for the GEM Shared Task at INLG'24 for being an underrepresented language. We have utilized the XLSUM dataset and have manually summarized 1000 texts from a Swahili news classification dataset. To understand the baseline, we have tested abstractive summarizers (mT5_multilingual_XLSum, t5, mBART), and an extractive summarizer (based on the PageRank algorithm). However, our adopted system consists of an integrated extractive-abstractive model combining the Bert Extractive Summarizer with an abstractive summarizer (t5 or mBART). The integrated model overcomes the drawbacks of both the extractive and abstractive summarization systems and utilizes the benefits from both of them. Our Integrated extractive-abstractive (t5) system performed better than other systems and outperformed GPT-3.5 in the final evaluation.

1 Introduction

In sub-Saharan Africa, Swahili has been regarded as the most spoken language. It has been serving as the national language of Tanzania and Kenya and is also widely spoken in Uganda, Rwanda, Burundi, the Democratic Republic of Congo, and Comoros. It has been the only African language with an estimated 100 million speakers and has played an important role in East and Central Africa as a lingua franca (at Ohio University Swahili Language). Therefore, summarizing tasks in the Swahili language is crucial.

Summarization in the Swahili language has faced challenges because of its rich morphology, multiple dialects, and regional variations (as mentioned in Jerro, 2018). These variations have been important in understanding the context essential for producing relevant summaries. No study has proposed a Swahili-specific monolingual language model with culturally diverse data, mainly due to Swahili being a low-resource language(LRL) with limited data availability (Martin et al., 2022). Additionally, there has been limited research on Swahili summarization.

The primary goal of this paper has been to summarize the Swahili texts for the Generation, Evaluation, and Metrics (GEM) Workshop at the International Conference on Natural Language Generation (INLG'24). The dataset used in this workshop has been introduced by Davis, 2020 and consists of Swahili news classification texts along with their respective classes.

In recent years, automatic text summarization has gained popularity for its ability to summarize text efficiently, quickly, and accurately while maintaining context. It has been categorized into two classes: extractive summarizers and abstractive summarizers (Hahn and Mani, 2000).

Extractive Text Summarizers (ETS) use mathematical calculations to measure sentence similarity. From this sentence similarity, a similarity matrix is formed which is then converted into a graph. In the graph, sentences are nodes and similarity scores are edges. Finally, the summary is constructed from the sentences with the top scores. This can be problematic in some cases. If a text covers multiple topics, like sports and politics, the similarity score diminishes as the topic changes, and thus in the summary, both topics may not be present. Moreover, the highest-scoring sentences may cause redundancy. An Abstractive Text Summarizer (ATS) focuses on the salient

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 112–117
September 23 –27, 2024.

concepts in a text. It not only selects key pieces from the text but also presents these key concepts in a new way, thereby eliminating the redundancy problem often encountered with ETS. Additionally, ATS captures the essence of the text even with multiple topics. However, it is more complex than ETS and is typically implemented using LSTM, seq2seq model. A limitation of ATS is that it can only process up to a limited number of tokens as input and any tokens beyond this limit are truncated. As a result, valuable information may be lost. Thus, ATS is not fully beneficial for summarizing long texts that contain a large number of tokens.

A fusion of extractive and abstractive text summarizers can help by utilizing the strengths of both methods. In many texts of our dataset, the number of tokens has exceeded 512. At first, we have implemented an extractive summarizer that reduced the size of the text beyond 512 tokens, keeping all possibly important information. This slightly summarized text has been summarized again by the abstractive summarizer for further refinement. Slightly summarized texts containing fewer than 512 tokens have undergone direct processing by the abstractive summarizer without using the extractive method. This method has enabled the summarization of longer texts and has provided coherent and comprehensive summaries.

To achieve our goal, we have manually summarized 1000 texts from the provided Swahili news classification dataset. Next, We have combined the XLSUM dataset with our manually prepared summaries. After that, we evaluated three abstractive summarizers (mT5_multilingual_XLSum, t5-small, mBART-50), one extractive summarizer (based on the PageRank algorithm), and two integrated extractive-abstractive summarizes. In the integrated system, we have integrated the Bert Extractive Summarizer with some abstractive summarizers(t5-small, mBART-50).

During our comparative analysis, we have trained all the systems on the prepared dataset. The integrated extractive-abstractive summarizer system with the "t5-small" model emerged as the most effective, achieving the highest ROUGE scores. In the final evaluation, This system outpaced GPT-3.5 in the automatic evaluation report in Mille et al., 2024.

Our core contributions in this work include the manual summarization of 1,000 texts from the provided Swahili news classification dataset (Davis, 2020). Also, we have integrated a Bert Extractive Summarizer and an Abstractive Summarizer to ensure context-based summarization of the larger texts. Detailed information on implementation is available in the GitHub repository linked below- `https://github.com/Samia2001/CUET_SSTM_GEM24`.

2 Related Work

The effort of enabling computers to automatically generate summaries has been practiced extensively, due to its vast applications in the processing of natural languages. Previous works on automatic text summarization can be classified into two categories (Hahn and Mani, 2000). They are Extractive summarization and Abstractive summarization.

One of the pristine approaches in Extractive summarization has been found in Luhn, 1958. They have taken into account the frequency of words and their relative positions to rank the sentences. Graph-based algorithms have been used to introduce faster and more scalable extractive summarization approaches. TextRank (Mihalcea and Tarau, 2004) and PageRank (Page, 1998) are two basic and prominent graph-based extractive summarizers. Later on, many other graph-based algorithms have been developed based on these two algorithms. Such as LexRank (Erkan and Radev, 2004) based on PageRank and TopicRank (Bougouin et al., 2013), PositionRank (Florescu and Caragea, 2017) based on TextRank.

Due to the lack of comprehensibility and rationality of Extractive summarization approaches (as mentioned in Saggion and Poibeau, 2013), Abstractive summarization has been introduced. CNN, RNN, LSTM-GRU and GAN-based approaches have been used frequently (Rekabdar et al., 2019, Yang et al., 2020). However, the ultimate improvement in summarization has been done by using Transformers. T5 (Raffel et al., 2020), BART (Lewis et al., 2019) etc. transformer architectures have been used to summarize texts. They have multilingual versions such as mT5 (Hasan et al., 2021) and mBART (Tang et al., 2020) which enables summarization in the Swahili language. Long text summarization has been a drawback of Abstractive summarization. To overcome this issue, integration of both the extractive and abstraction have been proposed in Wang et al., 2017.

3 Data

The given dataset (Davis, 2020) for this shared task contains a total of 23268 texts that have been collected from BBC News Swahili[1] and several other Tanzanian news websites. We have not used this dataset because it has been prepared for text classification rather than summarization. As a result, they don't contain a summary which is a must for training the system. Manually summarizing and training with such a large dataset would have needed a significant amount of time and resources.

We have utilized a different dataset XLSUM (introduced in Hasan et al., 2021) that contains summaries. This dataset includes 7,898 training, 987 development, and 987 test samples. These samples have been merged, resulting in a total of 9,872 samples. We have also used a custom dataset called SWAS[2] (Swahili Summarization) which was taken from the dataset provided by Davis, 2020. We have taken the first 1,000 texts and generated summaries with GPT-4 ensuring understandability, compactness, grammaticality, coherence, faithfulness, and saliency. SWAS dataset and XLSUM samples together have yielded a total dataset of 10,872 samples which we have used for training and evaluating our system.

The merged dataset has been shuffled. After shuffling, 1,000 samples have been split as the validation set, and the remaining 9,872 samples have been used for training.

4 System

In the GEM 2024 shared task, we participated in subtask 1 of the Summarization task, which is an unimodal task. The input text document and the generated summary both are in the Swahili language.

4.1 Data Preprocessing

During summarization, texts have been used as inputs and summaries as outputs. As both have needed preprocessing and summaries are the labels, special care has been required during preprocessing. Thus, we have used different preprocessing functions for texts and summaries. In both cases, all the uppercases have been lowered. However, the removal of punctuation, digits, and stopwords has only been applied to the texts, not to the summaries. Though NLTK[3] has been the renowned method for the resource of stopwords, it has not contained the stopwords of Swahili. So we have used 74 stopwords found in a GitHub repository.[4]

4.2 Initial Experimentation

As the provided dataset does not contain summaries, we have initially approached extractive summarization to establish the baseline for this task. Figure 1 illustrates that this system has first read and tokenized input text into sentences. Next, each sentence has been vectorized based on word frequency, with stopwords eliminated. Cosine Distance has been used to calculate sentence similarity, forming a similarity matrix based on pairwise relationships between sentences. Then the similarity matrix has been converted into a graph, considering sentences as nodes and similarity scores as edges. The PageRank algorithm (as mentioned in Xing and Ghorbani, 2004) has been used to rank the sentences based on their centrality and importance. The top-ranked sentences have been chosen to construct the summary.

Afterward, we have implemented abstractive summarization on our processed dataset. For this, we have used transformer-based models mBART-50, mT5_multilingual_XLSum, and t5. The "mBART-50" (introduced in Tang et al., 2020) has been a multilingual sequence-to-sequence model that supports 50 languages, including Swahili. We have used the "t5-small" checkpoint of the t5 model (introduced in Raffel et al., 2020), which con-

[1] www.bbc.com/swahili
[2] github.com/Samia2001/CUET_SSTM_GEM24
[3] www.nltk.org
[4] github.com/stopwords-iso/stopwords-sw

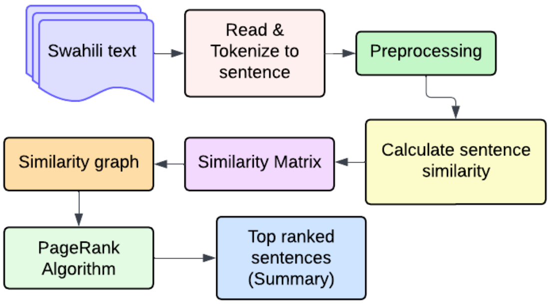


Figure 1: Initial Extractive summarization system.

tains about 60 million parameters. "mT5-multilingual-XLSum" has been an mT5 checkpoint (introduced in Hasan et al., 2021) fine-tuned on 45 languages of the XLSum dataset. We have evaluated this checkpoint to better understand the baseline scores for our dataset. Figure 2 illustrates the initial abstractive summarization system.

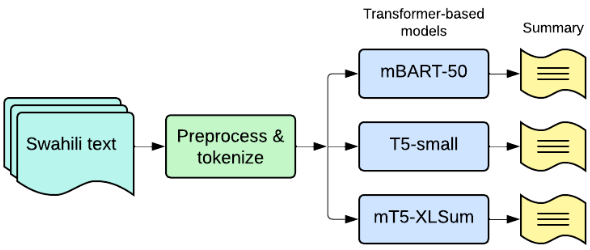


Figure 2: Initial Abstractive summarization system.

4.3 Overview of the Adopted Model

Our final adopted model has been the integration of both extractive and abstractive summarization systems. Our system takes texts of any length and outputs summaries in the Swahili language. Our dataset has contained very large texts, often exceeding 512 tokens, which has been the maximum input token limit of the transformer models. Extractive summarization has been used to shorten these very large texts (more than 512 tokens).

In this system, the input text has first been tokenized. Then, the token count has been checked. If the token count has exceeded 512, we have applied the "Bert Extractive Summarizer" tool available in Python[5] to reduce its token size to less than 512 tokens. This BERT

[5]pypi.org/project/bert-extractive-summarizer

Extractive Summarizer has ensured that the output is large enough to retain valuable information. Subsequently, we have applied a transformer-based model to produce a more precise and accurate summary. We have used the "t5-small" and "mBART-50" checkpoints with the Seq2SeqTrainer API. We haven't used the "mT5-XLSum" in the final adopted model, because this model is pre-trained on the "XL-SUM" dataset we used to train our system. We only used this model in section 4.2 to get a baseline for our task.

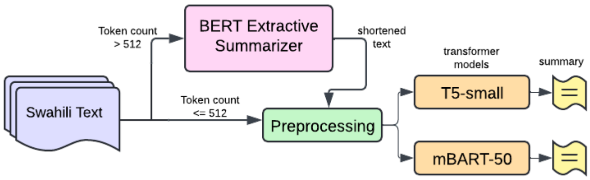


Figure 3: Integrated extractive and abstraction summarization system

5 Results and Analysis

5.1 Parameter Setting

To train the transformer model "t5-small", the parameters have been set as follows: dropout rate and attention rate have been set to 0.1, learning rate to 0.00005, training and evaluation batch size both set to 16, no weight decay and ran for 100 epoch with conditions to save the best model enabling early stopping. The patience for early stopping has been set to 3. It has run for 86 epochs before stopping. "mBART-50" benchmark has also been trained and its parameter setting was as follows: learning rate has been set to 0.00001, training and evaluation batch size both to 8, weight decay 0.1 and ran for 37 epoch.

5.2 Evaluation Metrics

There have been two types of evaluation for this task as mentioned in Mille et al., 2024. They are human evaluation and automatic evaluation. Understandability, faithfulness, saliency, grammaticality, coherence, and compactness of each generated summary have been checked in human evaluation. In the automatic evaluation, ROUGE scores (ROUGE-1 and ROUGE-2), BARTScore, and BERTScore have been evaluated.

5.3 Comparative Analysis

We have evaluated our systems with the validation set of our dataset. To ensure the significance of our evaluation result, we have split our dataset into 3 subsets after shuffling named SW-A, SW-B, and SW-C. The evaluation metrics we used for this evaluation have been only ROUGE scores (ROUGE-1 as R1, ROUGE-2 as R2, and ROUGE-L as RL). Table 2 presents the ROUGE scores of the systems used in our study. It clearly shows the Integrated Extractive-Abstractive system where "t5" has been used as the abstractive model, outperformed all other systems. Though the scores are very close, but 3 validation set's score proves the significance of the statistical difference. Thus we submitted this system (named as "CUET_SSTM") for final evaluation. Compared to GPT-3.5, "CUET_SSTM" performed better in ROUGE scores, equal scores in BERTScore, and lower scores in BARTScore.

System	Val Set	ROUGE Score		
		R1	R2	RL
Extractive	SW-A	0.06	0.01	0.05
	SW-B	0.07	0.02	0.06
	SW-C	0.04	0.01	0.03
Abstractive (mBART)	SW-A	0.14	0.03	0.1
	SW-B	0.14	0.03	0.1
	SW-C	0.1	0.02	0.08
Abstractive (t5)	SW-A	0.14	0.05	0.12
	SW-B	0.13	0.03	0.13
	SW-C	0.11	0.03	0.1
Abstractive (mT5-XLSUM)	SW-A	0.11	0.04	0.1
	SW-B	0.1	0.03	0.1
	SW-C	0.09	0.03	0.1
Integrated (t5)	SW-A	**0.16**	**0.06**	**0.15**
	SW-B	0.16	0.05	0.15
	SW-C	0.15	0.05	0.14
Integrated (mBART)	SW-A	0.14	0.04	0.12
	SW-B	0.14	0.05	0.13
	SW-C	0.13	0.04	0.11

Table 1: Performance of different systems on the validation subset

System	R1	R2	BART Score	BERT Score
GPT-3.5	27.12	10.42	-6.305	71.15
CUET_SSTM	**29.33**	**15.87**	-6.791	71.15

Table 2: Performance of our Integrated extractive-abstractive (t5) system in final evaluation in GEM'24.

5.4 Discussion

Our model produced very condensed summaries, typically 1-2 lines, due to the small labeled summaries in the XLSUM dataset used for training. To maintain consistency, the dataset we have created also contains brief summaries. XLSUM's labeled summaries don't ensure six key criteria required by the shared task—understandability, compactness, grammaticality, coherence, faithfulness, and salience. However, we've ensured these qualities in our manually created dataset, but it consists of only 1,000 entries. Additionally, the dataset we have used is relatively small (10,872 training samples), and the extractive summarizer we have used relies on cosine similarity, and does not always capture the full essence of longer texts. Our future work aims to produce summaries that accurately reflect and capture the essence of the original text. We also plan to expand our manually created dataset while ensuring it meets the six key criteria mentioned above.

6 Conclusion

Swahili summarization is challenging due to limited resources and no dedicated models. We manually summarized 1,000 texts from a Swahili news classification dataset and combined them with XLSUM's Swahili data. Using an extractive-abstract method, we applied a BERT-based summarizer for length reduction, followed by an abstractive T5-small model. Our system outperformed GPT-3.5 in R1 and R2 scores and matched its BERTScore, but GPT-3.5 outperformed our model in BART-score, particularly with highly condensed summaries.

Ethics Statement

While analyzing, preprocessing, and implementing the systems, we have ensured to keep the highest ethical standards. Our contribution will impact positively the development of a more sophisticated summarization system in the Swahili language by helping mass people.

References

OHIO University OHIO Center International Studies African Studies African Languages at Ohio University Swahili Language. Swahili. https://www.ohio.edu/cis/african/languages/swahili.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *IJCNLP*.

David Davis. 2020. Swahili: News classification dataset (0.1).

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*.

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the ACL (volume 1: long papers)*.

Udo Hahn and Inderjeet Mani. 2000. The challenges of automatic summarization. *Computer*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the ACL: ACL-IJCNLP 2021*. ACL.

Kyle Jerro. 2018. Linguistic complexity: A case study from swahili. *African linguistics on the prairie*.

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*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. 2022. Swahbert: Language model of swahili. In *Proceedings of the 2022 Conference of the North American Chapter of the ACL: Human Language Technologies*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Lawrence Page. 1998. The pagerank citation ranking: Bringing order to the web. technical report. *Stanford Digital Library Technologies Project, 1998*.

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*.

Banafsheh Rekabdar, Christos Mousas, and Bidyut Gupta. 2019. Generative adversarial network with policy gradient for text summarization. In *2019 IEEE 13th international conference on semantic computing (ICSC)*. IEEE.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. *Multi-source, multilingual information extraction and summarization*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. 2017. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE international congress on big data (BigData congress)*. IEEE.

Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004*. IEEE.

Min Yang, Chengming Li, Ying Shen, Qingyao Wu, Zhou Zhao, and Xiaojun Chen. 2020. Hierarchical human-like deep neural networks for abstractive text summarization. *IEEE Transactions on Neural Networks and Learning Systems*.

The LSG Challenge Workshop at INLG 2024: Prompting Techniques for Crafting Extended Narratives with LLMs

Aleksandr Boriskin[1], Daria Galimzianova[1]

[1]MTS AI / Moscow, Russia

Abstract

The task of generating long narratives using Large Language Models (LLMs) is a largely unexplored area within natural language processing (NLP). Although modern LLMs can handle up to 1 million tokens, ensuring coherence and control over long story generation is still a significant challenge. This paper investigates the use of summarization techniques to create extended narratives, specifically targeting long stories. We propose a special prompting scheme that segments the narrative into several parts and chapters, each generated iteratively with contextual information. Our approach is evaluated with GAPELMAPER, a sophisticated text coherence metric, for automatic evaluation to maintain the structural integrity of the generated stories. We also rely on human evaluation to assess the quality of the generated text. This research advances the development of tools for long story generation in NLP, highlighting both the potential and current limitations of LLMs in this field.

1 Introduction

Long story generation with LLMs is an underexplored topic in NLP. Most recent LLMs with wide context windows intuitively seem to be an appropriate tool for this task. However, in practice researchers often struggle to control the generation and keep it consistent (Kreminski and Martens, 2022).

We propose to use summarization and for LLMs to generate long stories of 40,000 words in length. Our approach does not require any fine-tuning and utilizes Llama 3 with 70b parameters with special prompting scheme. We score relevance, consistency, fluency and coherence of the text in human evaluation and GAPELMAPER in automatic evaluation on the LSG Challenge Task.

We make the code publicly available.[1]

[1]https://github.com/sashaboriskin/long_story_generation

Our pipeline can be summarized as follows:

1. Summary generation;
2. Chapter generation:
   (a) Generating the beginning of the chapter;
   (b) Generating the climax of the chapter until a certain length in characters is reached;
   (c) Generating the end of the chapter;
3. Merging the chapter with the whole book;
4. Summarizing the generated chapter to contextualize further chapter generations.

Detailed scheme of our pipeline can be found in Figure 1.

2 Related Work

Long story generation is a dynamic field of NLP with new approaches emerging quickly after the introduction of LLMs.

Co-authoring with LLMs has been suggested in (Wang et al., 2024). Storyverse, a system for human-driven story generation, leverages LLMs for character simulations. The plot is written by humans, while a language model is responsible for detailed story development. We propose to automate the process of plot creation as well, with specifically crafted role prompts for an instruction-tuned model.

Another interesting approach of using language models as co-writers is explored in (Zhao et al., 2023). Interleaved (generated with human help) stories are found to be less preferred by human readers than non-interleaved stories. We use these findings to construct a long narrative in a fully automatic way. Unlike the authors of this work, who only test their approach on commercial models, we deploy an open-source LLM, which allows for more flexibility in tuning generation parameters.

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 118–122
September 23 –27, 2024. ©2024 Association for Computational Linguistics

Iterative story planning with LLMs is presented in (Xie and Riedl, 2024). This study utilizes prompting techniques and relies on findings from psychology to construct the system. In attempts to automate the story generation further, the authors of (Venkatraman et al., 2024) develop a multi-LLM prompt-based approach where different models are responsible for various story components generation.

We draw upon these and other works to introduce our approach that leverages prompting and summarization techniques to generate long fiction stories with an open-source LLM.

3 Method

In the competition baseline, the number of book parts (6) and the number of chapters in each part (12) are hardcoded. Then, in a loop, Mixtral model generates an entire part of the book, providing context in the form of the book's plot (main characters, storyline, etc.).

We decided to continue to develop the idea of the baseline based on generation of the book components. The full pipeline consists of 2 parts - summary generation and generation of chapters in a loop with the transmission of context about previous events in the book via the system prompt.

Here are our sampling parameters for both parts in Table 1:

Parameter	Summary	Chapters
temperature	0.5	0.5
top_p	0.9	0.9
repetition_penalty	1.2	1.3
top_k	60	80

Table 1: Sampling Parameters for generating a summary of the book (table of contents) and generating chapters.

3.1 Summary generation

We use the Llama 3 70B Instruct model (AI@Meta, 2024) for generating long stories because it integrates well with vLLM, ensuring efficient deployment in our research. Llama is a group of open-source models, which provides flexibility in generation parameters. Also, it has shown strong performance in generating song lyrics from our personal experience, indicating its potential for creating coherent and engaging narratives.

Our pipeline starts by generating a short plot of the book by chapters. In general, there are 3

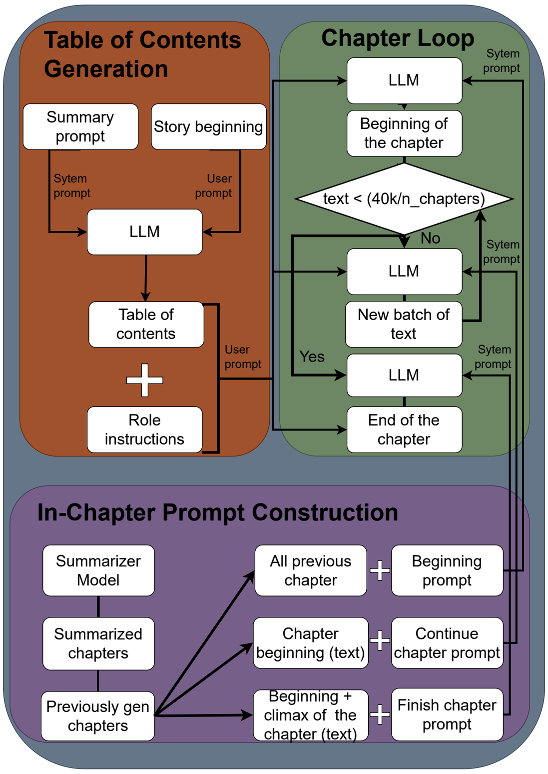


Figure 1: Pipeline scheme

compositional parts in the book: beginning, climax and outcome. We use instruction that describes the style (style of the fan fiction creator, magic elements in the book, format of chapters, etc.) for summary generation as a system prompt (can be found in Appendix A.1.1) for Llama3 70b Instruct, and the beginning of the story as a user prompt. We extract the exact number of chapters from the resulting summary with regular expressions. In our solution, the resulting summary contains 13 chapters.

3.2 Chapter Generation

The number of chapters, extracted from the summary with regular expressions, is used to generate each chapter in a loop. Each chapter is an independent part of the book with its own context, that's why we generate them in a loop separately. Next, we use the generated summary as a user prompt and prompt 3 (can be found in Appendix A.1.3) as a system prompt and transfer the entire previous chapter to it (if we generate the first chapter, then we transfer the beginning of the story), as well as the entire book summarized by the MT5 Multilingual XLSum model (Hasan et al., 2021). We use this model because of its high performance and specific strengths in summarization, all of which are essential for achieving robust and comprehensive results in previous and current research.

After the beginning of the story is generated, we generate the climax. We generate chunks of text with the system prompt 4 (can be found in Appendix A.1.4) with the same context as in the beginning of the story, except we do not transmit the entire previous chapter, but the current chapter. We do this iteratively until the length of the resulting chapter exceeds

$$\frac{40000(words)}{\text{number\_of\_chapters}}$$

This ensures that we generate a book of more than 40,000 words.

After we have reached the desired chapter length, we use prompt 5 (can be found in Appendix A.1.5) as a system prompt and we ask the model to finish the chapter, still transmitting the summarized context of the entire book and the entire current chapter.

This approach produces a coherent book of at least 40,000 words with standard composition structure of the plot (beginning-climax-outcome).

A potential improvement of our pipeline can be a new generation of Llama 3.1 models (which was released after the deadline of the competition) with an expanded context window up to 128k tokens, instead of 8k tokens with the 3rd generation. This would make it possible to present as context, if not the entire book, then large chunks of the book (including the last few chapters), rather than summarized information about the entire book.

We also came across the problem of a small number of words within one generation of the model without noticeable hallucinations (about 300-400 words). We tried to do this by experimenting with the `min_new_tokens` and `max_new_tokens` parameters, but this led to even more hallucinations of the model. If we had figured out how to increase the number of words within one generation, it would probably greatly increase the human evaluation of the metric, because within one generation the model makes fewer logical mistakes.

Our approach could potentially be extended to other genres and lengths of fiction stories, enabling the creation of diverse narrative forms, from short stories to epic novels. The iterative summarization technique ensures that narratives remain coherent and contextually rich, making it suitable for various styles and structures. For example, in the realm of video games, our technique can enhance interactive storytelling experiences by generating dynamic narratives that adapt to player choices, creating immersive and personalized gameplay.

3.3 Deployment

We produced our solution on 4 H100 80 GB GPUs, 128 GB RAM and 12 CPU cores.

For faster inference we used the VLLM framework. (Kwon et al., 2023)

With these resources, the generation of 13 chapters takes about 46 minutes.

4 Metrics and Evaluation

4.1 Automatic Evaluation

Fot automatic evaluation we use GAPELMAPER metric.

GAPELMAPER (GloVe Autocorrelations Power/Exponential Law Mean Absolute Percentage Error Ratio) (Mikhaylovskiy, 2023; Mikhaylovskiy and Churilov, 2023) is a metric designed to assess text coherence based on the autocorrelation of embeddings. It helps determine whether the text is intrinsically structured or not, based on the decay patterns of the autocorrelations.

The mathematical formula for GAPELMAPER can be represented as:

$$\text{GAPELMAPER} = \frac{\text{MAPE}_{\text{power}}}{\text{MAPE}_{\text{exp}}}$$

Metrics for our submission and the given baseline can be found in 2.

Metric	Baseline	Our
Power Mape	0.1796	0.5205
Log Mape	0.3014	0.9777
Exp Mape	0.3118	0.5713
GAPELMAPER	0.5760	0.9112

Table 2: Automatic metrics.

4.2 Human evaluation

The human evaluation metrics are texts rates across four dimensions: relevance (of topics in the text to the expected ones), consistency (alignment between the parts of the text), fluency (quality of individual sentences) and coherence (quality of sequence of sentences). Each dimension is evaluated on a scale from 1 to 5.

The values of the human metrics averaged over all asessors can be found in Table 3.

Human eval	baseline	ours
Relevance	3.4	2.05
Consistency	3.5	3.6
Fluency	3.8	3
Coherence	3.37	2.57

Table 3: Metrics assigned by human assessors.

The assessors are students whose average age is 20 years. They all study at a linguistic faculty, which confirms their high level of English proficiency ranging from B2 to C1. They read fiction in English about once a month on average.

The assessors have also provided extended feedback on the generated story. Their main concerns about the text included the following points:

- Semantic repetitions: some events in the texts are repeated several times, which makes the plot less structured.
- Logical inconsistencies: the narration between the chapters is sometimes interrupted by the events that logically could not happen at this particular point. The linearity of the overall plot is perturbed.
- Style of text: generally, the story looks like a summary of the whole Harry Potter book series, which does not match the fan fiction genre.
- Multiple plot endings: the story features several potential endings, which disrupts the overall composition of the book.
- Internal chapter composition: within individual chapters, the composition and connections are well-structured.
- Understanding of the Harry Potter universe: the model demonstrates a good understanding of the Harry Potter universe and can effectively create and develop characters within this world.

5 Conclusion

Our exploration into the use of summarization techniques for long story generation with Large Language Models has revealed promising avenues and notable challenges. The iterative generation process, combined with an evaluation metric like GAPELMAPER, shows potential in producing coherent and structured extended narratives. However, the difficulty in maintaining narrative consistency and control over extensive text generation underscores the need for further refinement of these techniques. Future work should focus on enhancing the control mechanisms and coherence metrics to better harness the capabilities of LLMs for long-form storytelling. This study lays the groundwork for more advanced narrative generation frameworks, pushing the boundaries of what LLMs can achieve in story telling task.

References

AI@Meta. 2024. Llama 3 model card.

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*.

Max Kreminski and Chris Martens. 2022. Unmet creativity support needs in computationally supported creative writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 74–82.

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*.

Nikolay Mikhaylovskiy. 2023. Long story generation challenge. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 10–16, Prague, Czechia. Association for Computational Linguistics.

Nikolay Mikhaylovskiy and Ilya Churilov. 2023. Autocorrelations decay in texts and applicability limits of language models. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2023"*, volume 2023.

Saranya Venkatraman, Nafis Irtiza Tripto, and Dongwon Lee. 2024. Collabstory: Multi-llm collaborative story generation and authorship analysis. *arXiv preprint arXiv:2406.12665*.

Yi Wang, Qian Zhou, and David Ledo. 2024. Storyverse: Towards co-authoring dynamic plot with llm-based character simulation via narrative planning. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–4.

Kaige Xie and Mark Riedl. 2024. Creating suspenseful stories: Iterative planning with large language models. *arXiv preprint arXiv:2402.17119*.

Zoie Zhao, Sophie Song, Bridget Duah, Jamie Macbeth, Scott Carter, Monica P Van, Nayeli Suseth Bravo, Matthew Klenk, Kate Sick, and Alexandre LS Filipowicz. 2023. More human than human: Llm-generated narratives outperform human-llm interleaved narratives. In *Proceedings of the 15th Conference on Creativity and Cognition*, pages 368–370.

A Appendix

A.1 Prompts

A.1.1 Prompt 1 - summary

You are a popular fanfiction creator tasked with writing an extended piece for a Harry Potter fanfiction. You have to come up with an interesting plot. Follow the classical composition and include beginning, climax and outcome in this book. Write a detailed plan for each composition chapter. In the format of

Chapter 1: What happens in this chapter

Chapter 2: What happens in this chapter

...

It is important to display the following aspects in the plot:

Action and Conflict: Introduce conflicts and challenges that the characters must face. Whether it's a battle with dark forces, a personal dilemma, or a complex mystery, ensure there is plenty of action and tension to keep readers hooked.

Magical Elements: Highlight the magical aspects of the Harry Potter universe. Describe new spells, potions, magical creatures, and enchanted locations. Make magic an integral part of the plot and the characters' lives.

World-Building: Expand on the existing lore of the Harry Potter universe. Introduce new locations, traditions, and histories. Make the world feel alive and full of possibilities.

Do not use p.s., p.p.s. and exc.

The following text is the beginning of the first chapter of this book. Generate the summary according to this text.

A.1.2 Prompt 2 - summary heading

You are a popular fanfiction creator tasked with writing an new chapter for a Harry Potter fanfiction according to the summary below. Your text should be distinct yet cohesive, maintaining the original tone and style of the Harry Potter series.

Instructions:

1. The text should be for secondary school students.

2. Always narrate in the third person.

3. Ensure that text is rich in detail and narrative depth.

4. Avoid including any text outside of the story (e.g., meta comments, thank you notes, or personal addresses).

5. Write the text with no additional comments.

6. Use only English letters and Arabic numerals.

Here is summary of whole text:

A.1.3 Prompt 3 - Start of the chapter

Start generating the chapter `{chapter_n}` based on what happened before.

here's what happened in the whole book so far: `{book_summary}`

here's what happened in the previous chapter so far: `{previous_chapter}`

Start writing the text with no additional comments.

The structure of the begging is:

Chapter `{chapter_n}`. Name of the chapter.

Text of the chapter

A.1.4 Prompt 4 - Continue generating the chapter

Continue generating the chapter `{chapter_n}` based on what happened before.

here's what happened in the whole book so far: `{book_summary}`

here's what happened in the chapter so far: `{full_chapter_context}`

Start writing the text with no additional comments.

Do not write chapter and the name of the chapter.

Just continue writing the story.

A.1.5 Prompt 5 - Finish generating the chapter

Finish generating the chapter `{chapter_n}` based on what happened before.

here's what happened in the whole book so far: `{book_summary}`

here's what happened in the chapter so far: `{full_chapter_context}`

Start writing the text with no additional comments. Do not write chapter and the name of the chapter. Just finish writing the story.

A Report on LSG 2024: LLM Fine-Tuning for Fictional Stories Generation

Daria Seredina
National Research University Higher School of Economics, Saint-Petersburg, 194100
daseredina@edu.hse.ru

Abstract

Our methodology centers around fine-tuning a large language model (LLM), leveraging supervised learning to produce fictional text. Our model was trained on a dataset crafted from a collection of public domain books sourced from Project Gutenberg, which underwent thorough processing. The final fictional text was generated in response to a set of prompts provided in the baseline. Our approach was evaluated using a combination of automatic and human assessments, ensuring a comprehensive evaluation of our model's performance.

1 Introduction

The increasing capabilities of machine learning have paved the way for generating various types of content using LLMs. Prompt-engineering methods, such as those proposed by Sanh et al. (2021), have demonstrated potential in creating fictional texts, but still require human oversight to produce coherent and engaging narratives (Guan et al., 2022). To overcome this limitation, when participating in the shared task of human-like long story generation, LSG Challenge (Mikhaylovskiy, 2023) we decided to explore a hybrid approach, integrating fine-tuning and prompt-engineering techniques to enhance long story generation results.

We chose to fine-tune the Mistral-7B-Instruct-v0.2-GPTQ (Jiang et al., 2023) LLM, aiming to make plausible fictional text generation possible. We define the following traits of the plausible fictional text:

- Metaphors and expressive means, complicated vocabulary, emotional depth, stylistic sophistication.
- Compliance with literary and stylistic norms established in various literary genres and directions.
- Convincing characters, peculiar storylines and conflicts that can interest and capture the reader's attention.
- Individuality and originality of the text, which allows the reader to learn new facts, feel emotions and get a unique reading experience.

In this context, fine-tuning of the model allows us to create a more advanced and adapted system for generating a literary text that takes into account the peculiarities of the literary fantasy genre.

2 Pipeline

To participate in the shared task, we leveraged a fine-tuned model and a custom pipeline to generate a comprehensive Harry Potter fanfiction. Our pipeline employs a hierarchical approach to narrative generation, utilizing prompting to create a lengthy and engaging story. The pipeline consists of three key stages:

- We initiate the process by loading the provided prompt and using an untrained Mistral-7B-Instruct-v0.2-GPTQ model to generate a high-level outline of the narrative similarly to the LSG Challenge baseline (Migal et al., 2024) and previous work (Lee et al., 2024, Sun et al., 2022). This outline serves as a roadmap, defining

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 123–127
September 23 –27, 2024. ©2024 Association for Computational Linguistics

the most critical events that will unfold in the story, including the setup, climax, resolution, and conclusion.

- With the narrative skeleton in place, we generate a more detailed content outline, comprising chapters that correspond to each part of the story. Each chapter is accompanied by a concise plot description, outlining the main events that will happen.

- Once we have a satisfactory chapter outline, we utilize our fine-tuned model (Mistral-7B-Instruct-v0.2-GPTQ) to generate the actual narrative, bringing the story to life.

3 Fine-tuning Approach

For fine-tuning the model, we employed a hybrid approach that combines elements of Supervised Learning (SL) and Self-Supervised Learning (SSL). Specifically, our dataset consisted of pairs of input data (brief summaries of a book chunks) and their associated labels (original book excerpts), which the model used to adjust its parameters and improve its predictive accuracy. This supervised learning aspect allowed the model to learn the mapping between input data and target outputs. However, the task of generating text based on brief summaries does not provide explicit labels for every possible output. Instead, the model must learn to generate text by leveraging the internal relationships between the input data and the desired output, which is a characteristic of self-supervised learning. We chose this hybrid approach due to its effectiveness in optimizing model performance when a clear relationship between inputs and desired outputs can be established.

To achieve this, we constructed a dataset that conforms to the following structure:

- "Outline": a concise summary of the narrative that the model should utilize to generate a fictional text.

- "Reference text": an exemplary text that serves as a benchmark for a well-crafted fictional text.

- "Instruction": a specific prompt that guides the model on how to integrate the " outline" and "reference text" to produce a coherent output.

The "outline" was derived from the original text chunk and distilled into a condensed summary. Consequently, the primary objective for the model during fine-tuning was to reconstruct the original text chunk with fidelity.

4 Dataset

The dataset was constructed using a part of the vast digital collection of public domain books provided by Project Gutenberg (Project Gutenberg, 2016). This collection comprises a broad spectrum of classic literature. To create the dataset, we selected over 500 books from the Project Gutenberg collection, starting from the first available ID, while striving to retain only books containing fictional works and discarding non-fiction texts, as

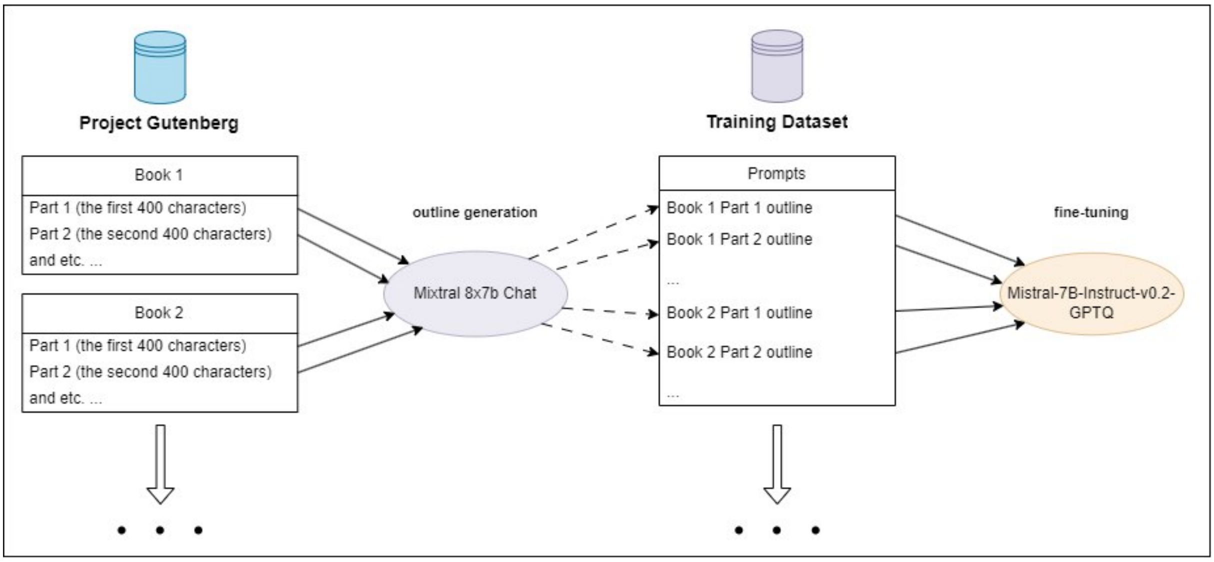


Figure 1: Dataset generation

our primary objective was to fine-tune a model capable of generating artistic text. These books were converted into text format and processed, enabling us to extract plotlines for generating concise summaries or “outlines” with Mixtral 8x7B model (Jiang et al., 2023). To prepare the data for model training, we transformed it into the required format by adding special tokens, the specific prompt, and the reference text, as follows:

“[INST] You are a famous writer and you are writing a book. Now you are going to write one chapter of your book according to this plot: In the beginning, <…>. Add as many descriptions, dialogues, feelings of the main characters and other things as possible. [/INST] Try to write your text as well as this one: *Reference text*.”

The final version of the dataset was used for training the model. The dataset creation process is shown in Figure 1.

5 Parameters

For fine-tuning, we utilized a LoraConfig, which includes the following parameters:

- r = 8 - This is a hyperparameter value that determines the vector size for each layer of the model.
- lora_alpha = 32 - Defines the learning scale for each layer of the model.
- target_modules = ["q_proj"] - This is a list of layers in the model that will be updated during fine-tuning. In this case, we only update the q_proj layer.
- lora_dropout = 0.05 - Determines the proportion of neurons to be disabled during training to prevent overfitting.
- task_type = "CAUSAL_LM" - This is the type of task for which we are training the model. In this case, we are training the model for text generation based on sequential prediction of the next token (causal language modeling).

6 Epochs testing

Before training the model, the language of a generated text was boring, primarily consisting of a list of actions performed by the main character. As the model was trained, the generated text became more engaging and nuanced, demonstrating the effectiveness of the chosen method.

After the first epoch of training, we obtained a model capable of generating more lively text based on the provided "plot" from a single sentence. The model was able to add new ideas to the provided plot and to provide the text with some special vibes that resembled a typical fairy tale or fantasy story:

Plot: "Harry Potter meets a magical spider. "

Epoch 1: "Harry Potter was walking through the forest when he stumbled upon a small, furry creature with eight legs. It was a spider, and it was very, very magical. <...> "Hello, Harry," said the spider. "I am a magical spider, and I can see into the future. <...>"

After the third epoch generated texts revealed to be written in a language rich in metaphors and complex vocabulary:

Epoch 3: "Harry Potter was walking through the forest when he suddenly felt a tickling sensation on his neck. He looked up and saw a small spider crawling up his hair. <...> It was a magical spider, and it had the power to grant wishes. <...> "

Epoch 4: "Harry Potter was walking through the forest one day when he came across a spider. It was a big, fat spider, with a glossy black body and a large, golden head. Harry was fascinated by the spider's appearance, and he was about to move on when he noticed that the spider was wearing a large, golden ring on its head. Harry was very curious about the ring, and he decided to investigate. <...> "

As the model continued to train, it became increasingly difficult for it to generate text based on a single, small sentence. In earlier epochs, the model excelled at "expanding the idea," while in later epochs, it focused on generating text strictly within the given "plot" scope, resulting in a more detailed storyline. However, when increasing the maximum number of tokens, later epochs began to duplicate the same idea presented in the plot throughout the text.

Text 4: "<...>Harry had never heard a spider speak before. He was afraid to say anything. "I am not a

```
spider," he said. "I am a boy."    "I
know that," the spider said. "But I am
a spider. I want to be
your friend."    Harry was still very
surprised. He did not know what to
say.     "I am not a boy," he said
again. "I am a man."    "I know that,"
the spider said. "But I am a spider.
I want to be your friend."
```

7 Evaluation

7.1 Automatic evaluation

The final text generated by the fine-tuned Mistral-7B-Instruct-v0.2-GPTQ was evaluated with GAPELMAPER Metric (Mikhaylovskiy, 2023; Mikhaylovskiy and Churilov, 2023). The evaluation results of the generated text are presented in Table 1, alongside the results calculated for well-known books' ("Don Quixote" and "The Adventures of Tom Sawyer") and one generated by S4 text's scores taken from Mikhaylovskiy and Churilov (2023). Mikhaylovskiy (2023) hypothesize that "GAPELMAPER less than 1 means that the autocorrelations decay according to a power law and the text is structured in a way. GAPELMAPER more than 1 means that the autocorrelations decay according to an exponential law and the text is unstructured". According to this statement the resulting text of our fine-tuned model exhibits a structured composition. Furthermore, the metrics obtained are similar to those achieved by "Don Quixote".

	Power law MAPE	Exp law MAPE	GAPEL-MAPER
Don Quixote	0.20	0.44	0.45
The Adventures of Tom Sawyer	0.21	0.55	0.38
S4 generated text	0.21	0.5	0.38
Mistral-7B-Instruct-v0.2-GPTQ Fine-tuned	0.17	0.402	0.44

Table 1: Automatic Evaluation Results

7.2 Human evaluation

The resulting fanfic was also subjected to human evaluation, as part of a blind assessment conducted by linguistics students. The participants were presented with a selection of anonymous fanfics, some of which were genuine and others generated by neural networks, without knowledge of their origin. After reading the texts, they completed a questionnaire, the results of which are presented in Table 2. Based on these scores, it can be inferred that the generated text demonstrates a satisfactory level of coherence, although articulating its core idea proves somewhat challenging.

From a language perspective, the text had some repetition, which suggested a limited vocabulary. However, the text's sentence structure was more complex, with features like parenthetical phrases, subordinate clauses, and participial phrases. The text also used common metaphors, comparisons, and oxymorons, but didn't go beyond these familiar expressions. The chapters showed a high degree of narrative repetition, leading one evaluator to suggest that the text was not written by a human. However, if this repetition is set aside, the evaluator believed that the writing style was typical of a young adult author who is a fan of the Harry Potter books.

Metric	Score
Correlation between the fanfic title and its content	3.25
Compatibility of chapter and sub-chapter titles with the overall style of the text	3.2
The strength of the stylistic connection between all the elements of the text	2.6
The pace of the plot	1.8
Word repetitions	2.6
Text composition	2.8
General idea of the text	3.2

Table 2: Artistic Quality Assessment Results

8. Results

As a result of our extensive research and experimentation, we were able to successfully fine-tune the model, testing it at various training epochs and gathering valuable insights into the aspects that require attention when fine-tuning it. One of the key findings from our investigation was that our fine-tuned model currently lacks the ability to make smooth transitions from one piece of text to another, specifically from one chapter to another. This limitation resulted in the appearance of sudden and unexpected plot twists, as well as the repetition of similar scenarios in adjacent chapters, which was readily apparent to our informants.

Despite this limitation, we were also able to generate a long and cohesive artistic text, which exhibited a certain level of structural quality, as measured by the results of our metrics (score of 0.44). Notably, this text featured a logical beginning and end, demonstrating a clear narrative arc. This achievement is significant, as it suggests that our fine-tuned model is capable of producing texts that are not only coherent but also engaging and well-structured.

Our research highlights the importance of addressing the issue of smooth transitions between chapters, as this is a critical aspect of creating a compelling and immersive narrative. By refining our model to better handle these transitions, we can potentially improve the overall quality and coherence of the generated texts.

Acknowledgements

The authors are grateful to their colleagues at NTR Labs ML division and classmates at HSE for the discussions and support. Our earlier discussions with Nikolay Mikhaylovskiy enabled us to successfully implement our idea and achieve our current result. We are also extremely grateful to Anastasia Kolmogorova for insightful discussions and valuable guidance throughout this process.

References

Jian Guan, Zhuoer Feng, Yamei Chen, Ruilin He, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2022. *LOT: A Story-Centric Benchmark for Evaluating Chinese Long Text Understanding and Generation.* Transactions of the Association for Computational Linguistics, 10:434–451.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. *Mixtral of Experts*. arXiv: 2401.04088

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed. 2023. *"Mistral 7b",* arXiv: 2310.06825

Yukyung Lee, Soonwon Ka, Bokyung Son, Pilsung Kang, Jaewook Kang. 2024. *Navigating the Path of Writing: Outline-guided Text Generation with Large Language Models,* Korea University, Naver AX Center, arXiv:2404.13919v1

Aleksandr Migal, Daria Seredina, Lyudmila Telnina, Nikita Nazarov, Anastasia Kolmogorova, Nikolay Mikhaylovskiy. *Overview of Long Story Generation Challenge (LSGC) at INLG 2024*. In Proceedings of the 17th International Natural Language Generation Conference, Tokyo, Japan

Nikolay Mikhaylovskiy. 2023. *Long Story Generation Challenge*. In Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges, pages 10–16, Prague, Czechia. Association for Computational Linguistics.

Nikolay Mikhaylovskiy and Ilya Churilov. 2023. *Autocorrelations Decay in Texts and Applicability Limits of Language Models*. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2023"

Project Gutenberg. (n.d.). Retrieved February 21, 2024, from www.gutenberg.org.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, et al. 2021. *Multitask Prompted Training Enables Zero-Shot Task Generalization.* ICLR 2022.

Xiaofei Sun, Zijun Sun, Yuxian Meng, Jiwei Li, Chun Fan. 2022. *Summarize, Outline, and Elaborate: Long-Text Generation via Hierarchical Supervision from Extractive Summaries.*, arXiv:2010.07074v2

Author Index