Volume 38 — Number 2 — 2025 — ISSN 2190-6858

# Journal for Language Technology and Computational Linguistics

# Special Issue on LLM Fails – Failed experiments with generative AI and what we can learn from them

Edited by Ngoc Duyen Tanja Tu, Annelen Brunner and Christian Lang

Gesellschaft für Sprachtechnologie und Computerlinguistik

# Imprint

**Editor**
Christian Wartena
Publication supported by the Gesellschaft für Sprachtechnologie und Computerlinguistik (GSCL)

**Board of Directors**
Committee and Advisory Board of the GSCL

**Current issue**
Volume 38 – 2025 – Number 2

**Address**
Christian Wartena
Hochschule Hannover
Expo Plaza 12
D-30539 Hannover
info@jlcl.org

**ISSN**
2190-6858

**Publication**
Mostly 2 issues per annum
Publication only electronically on jlcl.org

# Contents

Ngoc Duyen Tanja Tu / Annelen Brunner / Christian Lang

# Editorial

Failed experiments typically have no place in scientific discourse; they are discarded and not published. We believe that this practice results in a loss of potential knowledge gain. A systematic reflection on the causes of failures allows for the critical examination and/or improvement of methods used. Furthermore, when previously failed experiments are repeated and subsequently succeed, progress can be explicitly determined. From the perspective of methodological reflection, the discussion and documentation of failures thus provide added value for the scientific community. This is particularly true in a field like research on and with generative artificial intelligence (AI), which lacks a long-standing tradition and in which best practices are still in the process of being established.

This JLCL special issue focuses on linguistic and NLP experiments with generative AI that did not yield the desired results. All papers explore the extent in which their failed experiment can contribute to knowledge gain regarding the work with generative AI.

The first three papers test LLMs for various annotation and information extraction tasks.

**Elena Leitner and Georg Rehm** systematically evaluate LLMs on several classification tasks for German social media texts. In this context, they compare different fine-tuning and prompting techniques and point out weaknesses.

**Barbara Heinisch** describes challenges in the use of LLMs in terminology work, an area in which accuracy and replicability are of particular importance. She advocates a selective application of LLMs in terminology work, emphasizing the importance of evaluating their appropriateness for specific tasks rather than using them indiscriminately.

**Elena Volkanovska** tackles the question on how to systematically deal with LLM specific errors. She proposes an error classification framework complementary to established performance metrics for NER classifiers that accounts for additional possible outcomes in a few-shot, LLM-based NER task.

The following three papers focus on the struggles of LLMs when presented with tasks that require complex semantic analysis.

**Yanming Li and Meaghan Fowlie** test several prompting strategies for four GPT-models to perform Abstract Meaning Representation (AMR) parsing on natural language sentences and find the performance worse than that of state-of-the-art AMR parsers.

**Natalia Skachkova, Simon Ostermann, Josef van Genabith and Bernd Kiefer** investigate the ability of different (L)LMs in bridging generation. They challenge the models with two tasks: (1) generate texts containing bridging and (2) fill in missing (bridging) spans.

**Sebastian Reimann** and **Tatjana Scheffler** provide a series of zero- and few-shot experiments on the detection of linguistic metaphors and specifically on extended metaphors with LLaMa and GPT models.

The final two papers approach the topic from a more meta perspective.

**John David Storment** focuses on the ability of LLMs to provide linguistic acceptability judgments and shows that they struggle with texts that use emojis as morpho-grammatical components.

Finally, the contribution by **Simon Münker** widens the scope to LLMs as social agents: He investigates how personalized LLMs align with human responses on the Moral Foundation Theory Questionnaire. His results suggest that LLMs struggle to coherently represent ideologies, cautioning against using them to simulate social interactions.

We would like to thank the authors for their contributions, which enabled us to compile this thematically rich special issue. We also thank the reviewers for their thorough feedback. Last but not least, we want to thank the editor of the Journal for Language Technology and Computational Linguistics for his support in putting together this special issue. We wish the reader a pleasant and engaging reading experience!

The guest editors,
Ngoc Duyen Tanja Tu, Annelen Brunner and Christian Lang

**Figure 1:** LLMfails llama mascot - generated with GPT-4o

# Reviewer index

Matthias Aßenmacher
Department of Statistics, LMU Munich
Munich Center for Machine Learning (MCML)
matthias@stat.uni-muenchen.de

Louis Cotgrove
Department of Lexical Studies
Leibniz Institute for the German Language
cotgrove@ids-mannheim.de

Nils Diewald
Department of Digital Linguistics
Leibniz Institute for the German Language
diewald@ids-mannheim.de

Keli Du
Trier Center for Digital Humanities
University of Trier
duk@uni-trier.de

Jennifer Ecker
Department of Digital Linguistics
Leibniz Institute for the German Language
ecker@ids-mannheim.de

Tim Feldmüller
Department of Digital Linguistics
Leibniz Institute for the German Language
feldmueller@ids-mannheim.de

Leonard Konle
Lehrstuhl für Computerphilologie
University of Würzburg
leonard.konle@uni-wuerzburg.de

Marc Kupietz
Department of Digital Linguistics
Leibniz Institute for the German Language
kupietz@ids-mannheim.de

Mark-Christoph Müller
Department of Pragmatics
Leibniz Institute for the German Language
mark-christoph.mueller@ids-mannheim.de

Peter Meyer
Department of Lexical Studies
Leibniz Institute for the German Language
meyer@ids-mannheim.de

Nina C. Rastinger
Literary & Textual Studies
Austrian Centre for Digital Humanities and Cultural Heritage
ninaclaudia.rastinger@oeaw.ac.at

Ines Rehbein
Data & Web Science Group
University of Mannheim
rehbein@uni-mannheim.de

Nils Reiter
Department of Digital Humanities
University of Cologne
nils.reiter@uni-koeln.de

Jan Oliver Rüdiger
Department of Lexical Studies
Leibniz Institute for the German Language
ruediger@ids-mannheim.de

Josef Ruppenhofer
Center of Advanced Technology for Assisted Learning and Predictive Analytics
FernUniversität in Hagen
josef.ruppenhofer@fernuni-hagen.de

Roman Schneider
Department of Grammar
Leibniz Institute for the German Language
schneider@ids-mannheim.de

Martina Scholger
Department of Digital Humanities
University of Graz
martina.scholger@uni-graz.at

Pia Schwarz
Department of Digital Linguistics
Leibniz Institute for the German Language
schwarz@ids-mannheim.de

Karolina Suchowolec
Institute of Translation and Multilingual Communication
TH Köln - University of Applied Sciences
karolina.suchowolec@th-koeln.de

Sascha Wolfer
Department of Lexical Studies
Leibniz Institute for the German Language
wolfer@ids-mannheim.de

Elena Leitner, Georg Rehm

# Exploring the Limits of LLMs for German Text Classification: Prompting and Fine-tuning Strategies Across Small and Medium-sized Datasets


## Abstract

Large Language Models (LLMs) are highly capable, state-of-the-art technologies and widely used as text classifiers for various NLP tasks, including sentiment analysis, topic classification, legal document analysis, etc. In this paper, we present a systematic analysis of the performance of LLMs as text classifiers using five German datasets from social media across 13 different tasks. We investigate zero- (ZSC) and few-shot classification (FSC) approaches with multiple LLMs and provide a comparative analysis with fine-tuned models based on Llama-3.2, EuroLLM, Teuken and BübleLM. We concentrate on investigating the limits of LLMs and on accurately describing our findings and overall challenges.


## 1 Introduction

Large Language Models (LLMs) have had a global impact, revolutionising numerous fields and sectors. LLMs leverage very large datasets and advanced architectures, resulting in the ability to process even complex linguistic phenomena. It is generally accepted that they have improved accuracy compared to previous smaller Language Models (LMs), leading to more effective and efficient solutions for various NLP tasks. LLMs have been adopted as text classifiers that demonstrate competitive performance using zero- and few-shot strategies and also fine-tuning, e. g., in English (Pan, García-Díaz, & Valencia-García, 2024; Wang, Pang, Lin, & Zhu, 2024). For German, Münker, Kugler, and Rettinger (2024) test LLMs in an annotation task on Twitter data and report results that are comparable with BERT. Many other studies on social media data demonstrate strong capabilities of LLMs to identify hate speech and offensive language (Bauer, Preisig, & Volk, 2024; He et al., 2024; Zampieri, Rosenthal, Nakov, Dmonte, & Ranasinghe, 2023).

How well do LLMs work when it comes to the German language? To address this question, we investigate the performance of several LLMs on five German datasets using different prompting as well as fine-tuning techniques. Our goal is to find the best solution for binary or multi-class classification with a minimal number of examples with unbalanced class distribution in the data, as well as to investigate why experiments failed and how to improve performance. This includes the following tasks:

- Analysing failures and successes for zero-shot and few-shot prompting approaches as well as for fine-tuning with selected LLMs;
- Analysing the performance of LLMs on selected datasets with regard to the size and distribution of classes;
- Analysing model limitations on a specific linguistic phenomenon presented in selected data (e. g., toxic, offensive or hateful language in social media) as well as in general for the German language.

Our code, results as well as a detailed description of the conducted experiments are available on GitHub.[1]

## 2 Experiments

**Learning Approaches** To answer the question which approach is better suited for text classification depending on the size of data and number of classes, we utilised zero- and 8-shot prompting as well as parameter-efficient fine-tuning using QLoRa (Dettmers, Pagnoni, Holtzman, & Zettlemoyer, 2023). Figure 1 shows a prompt that includes an instruction to classify the text, definitions of classes, and a question. The question for a task was formulated simply, e. g., “Does the text contain any form of offensive language?” Since the LLM does not know which classes we assume to exist, we added definitions. In FSC, for each label, eight random examples were inserted; we used a fixed random seed to ensure reproducibility. To avoid a detailed answer and to get only a category name, we instructed the LLM to answer with one word and to use a category from the list as an answer. Since prompting in the ‘native’ language enhances LLM comprehension (He et al., 2024), the prompt was formulated in German.

> Please classify the text. The categories are defined as follows: {DEFINITIONS}
>
> Here are a couple of examples of categories assigned by experts: {EXAMPLES}*
>
> {QUESTION} Please answer with one word and use a category from this list as an answer: {CATEGORIES}
>
> Text: {SENTENCE}
>
> Answer: {ANSWER}

**Figure 1:** Prompting template (translated to English)

For the fine-tuning experiments, we utilised the available train and test sets. We also created validation sets using examples from the train sets for hyperparameter tuning. Due to the different sizes of each dataset, the different numbers and also distributions of classes, we used several hyperparameters to improve performance. However, we observed that when good results were achieved in some tasks, overfitting occurred on

[1] `https://github.com/elenanereiss/Limits-of-LLMs-for-German-Text-Classification`

other tasks with the same hyperparameters. Thus, to avoid overfitting but to evaluate the models on equal terms, we applied the early stopping technique and set other hyperparameters to default.

**Datasets** We focus on five small and medium-sized datasets from social media (covering a total of 13 tasks) with different granularities of annotations and unbalanced class distributions, developed for the German language (see Table 1). A detailed overview of the tasks can be found in Appendix A.

| Dataset | Citation | Tasks | Size |
|---|---|---|---|
| German COVID-19 Twitter | [submitted] | informativeness, topic, credibility | 643 |
| German Speech Acts | Plakidis and Rehm (2022) | coarse and fine-grained classification | 1959 |
| HASOC 2020 | Mandl, Modha, Kumar M, and Chakravarthi (2021) | coarse and fine-grained classification | 2899 |
| GermEval 2019 | Struß, Siegel, Ruppenhofer, Wiegand, and Klenner (2019) | coarse and fine-grained classification | 7026 |
| | | implicit/explicit offensive language | 2888 |
| GermEval 2021 | Risch, Stoll, Wilms, and Wiegand (2021) | toxic, engaging, fact claiming | 4188 |

**Table 1:** Overview of German datasets.

**Models** We use recent non-European, European and German LLMs such as multilingual Llama 3.2-3B (Grattafiori et al., 2024), European EuroLLM-9B (Martins et al., 2024) and Teuken-7B (Ali et al., 2024) as well as German BübleLM. For prompting we use instruction-tuned models and for fine-tuning the pre-trained base models. Further details can be found in Appendix B.

**Statement on Possible Data Contamination** We would like to state explicitly that there is a lack of information regarding the training data of the LLMs we experiment with. Their training data may contain training and test sets from the datasets selected in our evaluation study. As reported by Balloccu, Schmidtová, Lango, and Dusek (2024); Samuel, Zhou, and Zou (2025), when data contamination occurs, through memorisation instead of true generalisation, it can lead to inflated evaluation scores.

## 3 Findings and Challenges

We conducted 78 prompting and 52 fine-tuning experiments in total. All major challenges we faced occurred during prompting. The first challenge was to ensure that an instruction-tuned LLM returns only a class for a given text. During ZSC, in many cases, we received one or more sentences with an explanation of the class. We tried several variants of the prompts; it worked well when we explicitly instructed the LLM to respond with one word and use a class from the list for its answer. Teuken has shown the best performance – on average 99% of answers were one word. With Llama 3.2 we got about 96% and with EuroLLM 79%. During FSC, the rate changed (Teuken: 99.9%, EuroLLM: 91%, Llama 3.2: 85.2%).

We collected the answers from the LLMs and systematised the limitations. In general, we observe the following types of output:

- The answer contained a valid class label: (i) as a word, (ii) in one or more sentences, (iii) translated into German as a word or in one or more sentences (mostly by Teuken for the classes “OTHER”, “OFFENSE”, “Risk_Reduction” and “Case_Report”).
- While the text was classified, no class label was provided (happened often during FSC with LLama 3.2. – “YES”, “NO”, etc.).
- The text was not classified: (i) due to a lack of context or (ii) due to offensive content (e. g., using Llama 3.2 on all datasets except COVID-19 Twitter).
- Hallucinations that were (i) similar to a predefined class, e. g., “OPFN” instead of “OFFN” or “GGovernm_Decisions” instead of “Governm_Decisions” (this behaviour was mostly observed with EuroLLM), or (ii) random words (“WHO”, “Zombies”, etc.)

Due to the number of tasks, the second challenge was to filter out class labels from sentences. To get a valid predicted label, we tokenised each output and compared each token with the predefined classes. In cases where we found multiple valid class labels in an answer, we were unable to assign one class automatically, and left these answers unchanged; German translations were mapped to corresponding classes.

Some of our experiments failed technically. Originally, we planned to also test the German LLM LLäMmlein (Pfister, Wunderle, & Hotho, 2024). Unfortunately, we were unable to get an answer from various instruction-tuned models in the form of a class during prompting. Due to this limitation, it was not possible to manually edit each output and filter out a category. Fine-tuning a base model also failed. For LLäMmlein, we got an error message during the initialisation of the tokenizer that we have not been able to fix, which is why we decided to exclude this LLM from our experiments.

Experimenting with $n$-shot prompting, we found that Teuken and EuroLLM were already working to capacity at 10-shot. EuroLLM began to hallucinate when the maximum input length was exceeded. This is why we reduced the number of examples in FSC to 8, i. e., many-shot classification with 100 examples was not tested. However, we have done some test runs with Llama 3.2, which allows 128,000 input tokens. Already with the first tasks, we noticed that all metrics decreased. On the HASOC 2020 and GermEval 2019 datasets, the $F_1$-scores were even worse than in ZSC, i. e., around 0.18-0.29 points. We see this as evidence that the use of more examples does not necessarily result in better performance.

## 4 Results and Discussion

The results of the prompting and fine-tuning experiments are shown in Table 2. The fine-tuned LLMs achieved the best results in all tasks. A big difference between prompting and fine-tuning can be found in the tasks with fine-grained classes. $F_1$-scores doubled for almost all LLMs. The $F_1$-scores on the Speech Acts Dataset (coarse) using Llama

| Dataset | and Task | A | Llama 3.2 p | Llama 3.2 r | Llama 3.2 f1 | EuroLLM p | EuroLLM r | EuroLLM f1 | Teuken p | Teuken r | Teuken f1 | BübleLM p | BübleLM r | BübleLM f1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COVID-19 Twitter | | zs | .84 | .61 | .59 | .53 | .40 | .39 | .67 | .53 | .49 | – | – | – |
| | **informativeness** | fs | .50 | .57 | .52 | .74 | .60 | .64 | .68 | .63 | .65 | – | – | – |
| | 3 classes | t | .70 | .73 | .70 | .77 | .79 | **.78** | .65 | .68 | .65 | .73 | .72 | .72 |
| | | zs | .23 | .23 | .18 | .31 | .28 | .26 | .46 | .37 | .29 | – | – | – |
| | **topic** | fs | .51 | .44 | .40 | .24 | .26 | .18 | .62 | .38 | .31 | – | – | – |
| | 6 classes | t | .71 | .66 | **.67** | .52 | .53 | .50 | .54 | .61 | .56 | .61 | .57 | .58 |
| | | zs | .41 | .45 | .18 | .34 | .18 | .23 | .41 | .39 | .38 | – | – | – |
| | **credibility** | fs | .54 | .36 | .41 | .45 | .63 | .39 | .56 | .51 | .44 | – | – | – |
| | 3 classes | t | .50 | .52 | .50 | .54 | .56 | **.54** | .47 | .48 | .47 | .51 | .52 | .51 |
| Speech Acts | | zs | .31 | .22 | .15 | .20 | .24 | .18 | .25 | .19 | .12 | – | – | – |
| | **coarse** | fs | .42 | .32 | .29 | .21 | .22 | .18 | .19 | .24 | .19 | – | – | – |
| | 6 classes | t | .64 | .67 | **.65** | .69 | .56 | .59 | .58 | .60 | .56 | .48 | .54 | .49 |
| | | zs | .13 | .11 | .10 | .15 | .13 | .12 | .03 | .11 | .03 | – | – | – |
| | **fine** | fs | .28 | .20 | .17 | .10 | .14 | .08 | .11 | .17 | .10 | – | – | – |
| | 17 classes | t | .33 | .38 | .34 | .34 | .32 | .31 | .40 | .43 | **.39** | .27 | .31 | .28 |
| HASOC 2020 | | zs | .66 | .62 | .59 | .60 | .62 | .53 | .63 | .51 | .22 | – | – | – |
| | **coarse** | fs | .70 | .49 | .55 | .63 | .67 | .63 | .63 | .52 | .24 | – | – | – |
| | 2 classes | t | .76 | .81 | .78 | .75 | .79 | .76 | .78 | .81 | **.79** | .78 | .78 | .78 |
| | | zs | .38 | .38 | .29 | .33 | .33 | .21 | .26 | .27 | .07 | – | – | – |
| | **fine** | fs | .39 | .30 | .29 | .36 | .32 | .24 | .23 | .28 | .25 | – | – | – |
| | 4 classes | t | .49 | .58 | .50 | .48 | .56 | .51 | .46 | .59 | .48 | .49 | .54 | **.51** |
| GermEval 2019 | | zs | .66 | .62 | .56 | .65 | .65 | .57 | .16 | .50 | .24 | – | – | – |
| | **coarse** | fs | .65 | .50 | .34 | .68 | .70 | .66 | .16 | .50 | .24 | – | – | – |
| | 2 classes | t | .76 | .77 | .76 | .76 | .78 | **.76** | .73 | .76 | .74 | .73 | .76 | .74 |
| | | zs | .36 | .37 | .36 | .33 | .34 | .23 | .29 | .32 | .21 | – | – | – |
| | **fine** | fs | .41 | .26 | .28 | .33 | .31 | .23 | .40 | .29 | .11 | – | – | – |
| | 4 classes | t | .42 | .45 | .42 | .40 | .45 | .40 | .44 | .48 | **.44** | .40 | .46 | .41 |
| | | zs | .54 | .53 | .26 | .51 | .48 | .28 | .43 | .50 | .46 | – | – | – |
| | **offensive** | fs | .57 | .51 | .14 | .54 | .52 | .37 | .60 | .50 | .47 | – | – | – |
| | 2 classes | t | .68 | .76 | **.70** | .68 | .71 | .69 | .67 | .74 | .69 | .65 | .73 | .67 |
| GermEval 2021 | | zs | .61 | .60 | .60 | .57 | .48 | .40 | .62 | .53 | .37 | – | – | – |
| | **toxic** | fs | .59 | .60 | .59 | .55 | .54 | .54 | .62 | .56 | .44 | – | – | – |
| | 2 classes | t | .67 | .68 | .68 | .70 | .69 | .69 | .70 | .71 | **.70** | .68 | .66 | .67 |
| | | zs | .55 | .54 | .53 | .56 | .49 | .46 | .56 | .52 | .33 | – | – | – |
| | **engaging** | fs | .51 | .41 | .43 | .53 | .52 | .52 | .59 | .55 | .35 | – | – | – |
| | 2 classes | t | .66 | .68 | .67 | .66 | .68 | .67 | .64 | .66 | .64 | .67 | .68 | **.67** |
| | | zs | .63 | .61 | .53 | .57 | .48 | .34 | .67 | .50 | .26 | – | – | – |
| | **factClaiming** | fs | .61 | .62 | .59 | .58 | .57 | .58 | .60 | .55 | .39 | – | – | – |
| | 2 classes | t | .77 | .77 | **.77** | .75 | .76 | .76 | .72 | .72 | .72 | .72 | .71 | .71 |

**Table 2:** Precision, recall and macro $F_1$-score for zero-shot (zs), few-shot (fs) classification and fine-tuning (t) on the test set. The best $F_1$-scores are marked as follows: in one approach underlined, in both prompting approaches underlined twice, and in all approaches in **bold**.

3.2 rose drastically from 0.31 in FSC to 0.65 after fine-tuning. For binary classification, the values also improved, at least 0.1 points.

Comparing the prompting approaches, the LLMs show better performance in FSC than in ZSC. Only in four tasks, Llama 3.2 was better in ZSC. The instruction-tuned models were surprisingly good at the identification of offensive and toxic content (binary classification), scoring around 0.6 $F_1$ (already in ZSC) on the HASOC 2020, GermEval 2019 and 2021 datasets. However, when the number of classes increases to 4 (i. e., in the fine-grained tasks), the LLMs fail, and the $F_1$-scores are in the range of only 0.25-0.35.

Regarding the small-sized datasets, unexpectedly, the fine-tuned LLMs exhibit solid performance on the classification of informativeness and topic (COVID-19 Twitter Dataset) and of coarse-grained speech acts. However, for the identification of credibility (3 classes), we expected better results. As anticipated, the LLMs performed worst at the classification of 17 highly unbalanced fine-grained speech acts. Regarding the medium-sized datasets, the results with fine-grained classes must be described as moderate. Even the fine-tuned LLMs only reached a maximum of 0.51 $F_1$ on HASOC 2020 and 0.44 $F_1$ on GermEval 2019.

As far as the LLMs are concerned, it is impossible to generalise which of the models is superior. Depending on the task and approach, some LLMs provide comparable results, such as EuroLLM and Teuken at topic classification with zero-shot prompting or at informativeness classification with 8-shot. In some tasks, the differences are enormous and reached a gap of almost 0.2 $F_1$. In ZSC, Teuken had 0.38 $F_1$ on the COVID-19 Twitter dataset (credibility) and 0.46 $F_1$ on GermEval 2019 (offensive). In FSC on GermEval 2019 (coarse), EuroLLM achieved 0.66 $F_1$. In ZSC on GermEval 2021, Llama 3.2 had 0.6 $F_1$ (toxic) and 0.53 $F_1$ (fact claiming). However, as we can see from Table 2, Llama 3.2 often scored the best $F_1$ depending on the task and approach.

We can draw the following conclusions from the experiments and evaluation:

- Fine-tuning outperforms prompting and is better suited for small- and medium-sized datasets with fine-grained annotations.
- Prompting achieves good results when a task is well-known and defined as binary classification.
- Prompting with the use of examples exhibits better performance than zero-shot.
- Apart from the chosen approach, the LLMs fail on small-sized datasets with fine-grained annotations with only a few examples per class.

## 5 Conclusion

Across prompting and fine-tuning approaches, LLMs exhibit satisfactory performance as text classifiers for German. The scores decrease rapidly as the number of labels increases. The fine-tuned LLMs significantly outperform the instruction-tuned LLMs in a zero- and 8-shot prompting approach. Moreover, the instruction-tuned LLMs exhibit certain limitations and are challenging to use.

## References

Ali, M., Fromm, M., Thellmann, K., Ebert, J., Weber, A. A., Rutmann, R., … Flores-Herr, N. (2024). *Teuken-7B-Base & Teuken-7B-Instruct: Towards European LLMs.* Retrieved from `https://arxiv.org/abs/2410.03730`

Balloccu, S., Schmidtová, P., Lango, M., & Dusek, O. (2024, March). Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In Y. Graham & M. Purver (Eds.), *Proceedings of the 18th Conference of the*

*European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 67–93). St. Julian's, Malta: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2024.eacl-long.5/

Bauer, N., Preisig, M., & Volk, M. (2024). Offensiveness, Hate, Emotion and GPT: Benchmarking GPT3.5 and GPT4 as Classifiers on Twitter-specific Datasets. In *Workshop on Trolling, Aggression and Cyberbullying.* Retrieved from https://api.semanticscholar.org/CorpusID:269950799

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.* Retrieved from https://arxiv.org/abs/2305.14314

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... Ma, Z. (2024). *The Llama 3 Herd of Models.* eprint arXiv:2407.21783. Retrieved from https://arxiv.org/abs/2407.21783

He, J., Wang, L., Wang, J., Liu, Z., Na, H., Wang, Z., ... Chen, Q. (2024). Guardians of Discourse: Evaluating LLMs on Multilingual Offensive Language Detection. *ArXiv*, *abs/2410.15623*. Retrieved from https://api.semanticscholar.org/CorpusID:273501874

Mandl, T., Modha, S., Kumar M, A., & Chakravarthi, B. R. (2021). Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation* (p. 29–32). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3441501.3441517 doi: 10.1145/3441501.3441517

Martins, P. H., Fernandes, P., Alves, J., Guerreiro, N. M., Rei, R., Alves, D. M., ... Martins, A. F. T. (2024). *EuroLLM: Multilingual Language Models for Europe.* Retrieved from https://arxiv.org/abs/2409.16235

Münker, S., Kugler, K., & Rettinger, A. (2024). Zero-shot Prompt-based Classification: Topic Labeling in Times of Foundation Models in German Tweets. *ArXiv*, *abs/2406.18239*. Retrieved from https://api.semanticscholar.org/CorpusID:270737586

Pan, R., García-Díaz, J. A., & Valencia-García, R. (2024). Comparing Fine-Tuning, Zero and Few-Shot Strategies with Large Language Models in Hate Speech Detection in English. *Computer Modeling in Engineering & Sciences.* Retrieved from https://api.semanticscholar.org/CorpusID:269943133

Pfister, J., Wunderle, J., & Hotho, A. (2024). *LLäMmlein: Compact and Competitive German-Only Language Models from Scratch.* Retrieved from https://arxiv.org/abs/2411.11171

Plakidis, M., Leitner, E., & Rehm, G. (2025). Automated Speech Act Classification in Offensive German Language Tweets. *Traitement Automatique des Langues*, *65*(3). (Special Issue on Abusive Language Detection)

Plakidis, M., & Rehm, G. (2022, June). A Dataset of Offensive German Language Tweets Annotated for Speech Acts. In N. Calzolari et al. (Eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 4799–4807).

Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2022.lrec-1.513/

Remy, F., Delobelle, P., Avetisyan, H., Khabibullina, A., de Lhoneux, M., & Demeester, T. (2024). *Trans-Tokenization and Cross-lingual Vocabulary Transfers: Language Adaptation of LLMs for Low-Resource NLP.* Retrieved from https://arxiv.org/abs/2408.04303

Risch, J., Stoll, A., Wilms, L., & Wiegand, M. (2021, September). Overview of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments. In J. Risch, A. Stoll, L. Wilms, & M. Wiegand (Eds.), *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments* (pp. 1–12). Duesseldorf, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.germeval-1.1/

Samuel, V., Zhou, Y., & Zou, H. P. (2025, January). Towards Data Contamination Detection for Modern Large Language Models: Limitations, Inconsistencies, and Oracle Challenges. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 5058–5070). Abu Dhabi, UAE: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2025.coling-main.338/

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & Klenner, M. (2019). Overview of GermEval Task 2, 2019 Shared Task on the Identification of Offensive Language. In *Conference on Natural Language Processing.* Retrieved from https://api.semanticscholar.org/CorpusID:208334401

Wang, Z., Pang, Y., Lin, Y., & Zhu, X. (2024). *Adaptable and Reliable Text Classification using Large Language Models.* Retrieved from https://arxiv.org/abs/2405.10523

Zampieri, M., Rosenthal, S., Nakov, P., Dmonte, A. E., & Ranasinghe, T. (2023). OffensEval 2023: Offensive language identification in the age of Large Language Models. *Natural Language Engineering*, *29*, 1416 - 1435. Retrieved from https://api.semanticscholar.org/CorpusID:265659717

## A Task Overview

To evaluate the performance of LLMs, we selected five German datasets comprising 13 tasks. Table 3 lists the tasks and their definitions as well as illustrates the number of classes in a task and the minimum number of instances per task.

The German COVID-19 Twitter Dataset[2] is a novel credibility dataset consisting of 643 COVID-19-related texts extracted during the pandemic. Credibility is framed as informative and relevant content regarding a predefined set of topics and therefore each

[2]Due to X's content redistribution policy, the dataset is not published. A paper on the dataset is currently under review.

| Dataset | Task | Identification and classification of | No. | Min. |
|---|---|---|---|---|
| German COVID-19 Twitter Dataset | *Informativeness* | Informative content related to COVID-19 | 3 | 55 |
| | *Topic* | Topic-related content | 6 | 15 |
| | *Credibility* | Credible content related to COVID-19 | 3 | 10 |
| German Speech Acts Dataset | *Coarse* | Coarse-grained speech acts | 6 | 20 |
| | *Fine* | Fine-grained speech acts | 17 | 11 |
| HASOC 2020 | *Coarse* | Hate, offensive and profane content | 2 | 907 |
| | *Fine* | Hate, offensive and profane content | 4 | 170 |
| GermEval 2019 | *Coarse* | Offensive language | 2 | 2257 |
| | *Fine* | Offensive language | 4 | 263 |
| | *Offensive* | Explicit and implicit offensive language | 2 | 393 |
| GermEval 2021 | *Toxic* | Toxic comments on Facebook | 2 | 1472 |
| | *Engaging* | Engaging comments on Facebook | 2 | 1118 |
| | *Fact claiming* | Fact-claiming comments on Facebook | 2 | 1417 |

**Table 3:** Short description of the tasks. "No." means the number of classes in a task, "Min." means minimum number of instances per class in a dataset.

tweet is annotated for informativeness, topic and credibility. In the informativeness task, texts are classified into informative (*informative*), non-informative (*none*), and tweets that report personal experience (*personal_experience*). In the topic task, main COVID-19-related topics are *case report*, *consequences*, *governmental decisions*, *risk reduction*, and *vaccination*. Tweets that are not topic-related are marked as *none*. In the credibility task, tweets that have high or low credibility are classified as *credible* or *non-credible*. If it is not possible to decide from the text whether the content is credible or not, tweets are assigned the class *none*.

In the German Speech Acts Dataset[3] (Plakidis & Rehm, 2022), 1,959 sentences are annotated for six coarse- and 23 fine-grained speech acts. In the coarse-grained task, sentences shall be classified into following classes: *assertive*, *directive*, *expressive*, *commissive*, *unsure* and *other*. In the fine-grained task, assertive, directive, expressive, and commissive speech acts are split into fine-grained ones. Similarly to Plakidis, Leitner, and Rehm (2025), due to sparse occurrences in the dataset, we modified a few fine-grained classes reducing the number of classes from 23 to 17.

The HASOC 2020 Dataset for German[4] (Mandl et al., 2021) consists of 2,899 tweets including binary and fine-grained annotations regarding the classification of hate-offensiveness. In the coarse-grained task, the goal is to identify hate, offensive and profane content and classify tweets into two classes: hate and offensive ($HOF$) or non hate-offensive ($NOT$). In the fine-grained task, a distinction is made between texts that contain hate speech ($HATE$), offensive content ($OFFN$), profane words ($PRFN$) and texts that do not contain hate speech, profane, offensive content ($NOT$).

[3] `https://github.com/MelinaPl/speech-act-analysis`
[4] `https://hasocfire.github.io/hasoc/2020/call_for_participation.html`

The GermEval 2019 Dataset[5] (Struß et al., 2019) originates from a shared task on the identification of offensive language. As for HASOC 2020, the first task deals with the coarse-grained binary classification of offensive language (*OFFENSE* and *OTHER*), and the second task – with the fine-grained classification containing four classes (*PROFANITY*, *INSULT*, *ABUSE*, *OTHER*). The third task focuses on the classification of explicit and implicit offensive language using the classes *EXPLICIT* and *IMPLICIT*.

The GermEval 2021 Dataset[6] (Risch et al., 2021) consists of 4,188 Facebook posts and addresses three classification problems. The first task deals with the classification of toxic comments. The second task on the engaging comment classification focuses on rational, respectful, and reciprocal comments. Due to the spread of misinformation and fake news, the third task is dedicated to the classification of fact-claiming comments and conceived as a pre-processing step for manual fact-checking. All three tasks belong to binary classification and are marked with *1* and *0*.

## B Models Overview

In our experiments, we utilise several recently released LLMs such as Llama 3.2, EuroLLM, Teuken, LLäMmlein and BübleLM. Meta Llama 3.2-3B is a smaller and more efficient version of the Llama3 family (Grattafiori et al., 2024) trained on approx. 9 trillion tokens from publicly available online data. EuroLLM-9B (Martins et al., 2024) is an open-weight multilingual LLM trained on 4 trillion tokens divided across official European Union languages (and several additional languages). Teuken-7B (Ali et al., 2024) is also a European LLM developed by the OpenGPT-X project. It is trained on 4 trillion tokens where 60% of data is non-English (8.72% data is German). LLäMmlein (Pfister et al., 2024) is a German Tinyllama LM trained on only high-quality German data from RedPajama V2. The last model, BübleLM, is a small German LM based on Gemma-2-2B and trained on 3.5B tokens from the Occiglot-FineWeb project. The model is characterized by using trans-tokenization – a cross-lingual vocabulary transfer strategy – for language adaptation of LLMs (Remy et al., 2024).

The instruction-tuned LLMs used in prompting experiments are as follows:

- Llama-3.2-3B-Instruct[7]
- EuroLLM-9B-Instruct[8]
- Teuken-7B-instruct-research-v0.4[9]
- *several LLäMmlein chat models[10]

The pre-trained base LLMs used in fine-tuning experiments are as follows:

[5] `https://fz.h-da.de/iggsa/`
[6] `https://germeval2021toxic.github.io/SharedTask/`
[7] `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`
[8] `https://huggingface.co/utter-project/EuroLLM-9B-Instruct`
[9] `https://huggingface.co/openGPT-X/Teuken-7B-instruct-research-v0.4`
[10] `https://huggingface.co/collections/LSX-UniWue/llammlein-chat-preview-6734b15176c7f079f72a9291`

- Llama-3.2-3B[11]
- EuroLLM-9B[12]
- Teuken-7B-base-v0.6[13]
- *LLaMmlein_1B[14]
- bueble-lm-2b[15]

As reported in Section 3, the experiments with various instruction-tuned LLäMmlein models, as well as with the pre-trained base LLäMmlein model failed. Therefore, these LLMs are marked with * in both lists. BübleLM has no instruction-tuned version and was excluded from the prompting experiments.

[11] https://huggingface.co/meta-llama/Llama-3.2-3B
[12] https://huggingface.co/utter-project/EuroLLM-9B
[13] The model is available upon request.
[14] https://huggingface.co/LSX-UniWue/LLaMmlein_1B
[15] https://huggingface.co/flair/bueble-lm-2b

**Correspondence**

Elena Leitner

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
Berlin, Germany
elena.leitner@dfki.de

Georg Rehm

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
Berlin, Germany
georg.rehm@dfki.de

Barbara Heinisch

# Large language models for terminology work: A question of the right prompt?


## Abstract

Text-generative large language models (LLMs) offer promising possibilities for terminology work, including term extraction, definition creation and assessment of concept relations. This study examines the performance of ChatGPT, Perplexity and Microsoft CoPilot for conducting terminology work in the field of the Austrian and British higher education systems using strategic prompting frameworks. Despite efforts to refine prompts by specifying language variety and system context, the LLM outputs failed to reliably differentiate between the Austrian and German systems and fabricated terms. Factors such as the distribution of German-language training data, potential pivot translation via English and the lack of transparency in LLM training further complicated evaluation. Additionally, output variability across identical prompts highlights the unpredictability of LLM-generated terminology. The study underscores the importance of human expertise in evaluating LLM outputs, as inconsistencies may undermine the reliability of terminology derived from such models. Without domain-specific knowledge (encompassing both subject-matter expertise and familiarity with terminology principles) as well as LLM literacy, users are unable to critically assess the quality of LLM outputs in terminological contexts. Rather than indiscriminately applying LLMs to all aspects of terminology work, it is crucial to assess their suitability for specific tasks.


## 1 Introduction

Large Language Models (LLMs), capable of processing and generating human-like text, are transforming numerous professions (Eloundou, Manning, Mishkin, & Rock, 2023), including specialized translation and terminology management. Since LLMs utilize distinct approaches to generate and comprehend language, they fundamentally change the function of terminology (Massion, 2024) and the way how terminologists and specialized translators approach terminology work. Nevertheless, technology has long supported translation, from Computer-Assisted Translation (CAT) tools to terminology extraction software, corpus analysis and alignment tools (Rothwell, Moorkens, Fernández-Parra, Drugan, & Austermühl, 2023). While machine translation tools have long been a staple for translators, LLMs bring a new level of versatility, as they cannot only be used for translation per se but also for translation-related tasks, such as clarifying meaning, editing style, detecting errors or assuring quality (Siu, 2023).

Also terminologists use a wide range of specialized software for terminology work, including term extraction (Steurs, de Wachter, & de Malsche, 2015) or the management of terminology in terminological databases (Drewer & Schmitz, 2017). LLMs can enhance terminology work by efficiently extracting relevant terms (Hamm, 2025), generating definitions (Reineke, 2023) in context and by assessing language-variety-specific terminology (Heinisch, 2020). They can be used for finding equivalents across languages and terminology validation. Additionally, LLMs help terminologists establish relationships between concepts and verify proper terminology use, streamlining the overall process (Massion, 2024).

Terminology, defined as "set of designations [...] and concepts [...] belonging to one domain [...] or subject [...]" (*ISO 1087:2019 Terminology work and terminology science — Vocabulary*, 2019) is crucial in specialized communication because it ensures precision, consistency and clarity in communication. In specialized translation, effective terminology management improves translation efficiency and ensures the quality of the target text. Therefore, terminology management is pivotal, including the identification of terms and concepts, the extraction of candidate terms as well as the organization and validation of terms before storing and maintaining them in terminological databases (Steurs et al., 2015). One major challenge in multilingual terminology work is determining the equivalence between concepts (Hohnhold, 1990). To accurately interpret and use terminology, terminologists and translators must consider its domain, the system it belongs to and the specific context in which it appears. Thus, terminology is domain-specific, system-bound and context-dependent. Therefore, the question arises how LLMs perform in (selected) tasks aimed at (multilingual) terminology work.

## 2 Method

The theoretical framework of this paper is grounded in Wüster's General Theory of Terminology (Wüster, 1974), whose foundational principles continue to inform contemporary terminological practice as codified in the ISO standard (*ISO 704:2022 Terminology work — Principles and methods*, 2022) on terminology work — Principles and methods. This study forms part of the larger project *UniTermGPT: University Terminology in German in the Age of ChatGPT*, which explores how ChatGPT handles university terminology across selected German language varieties. Therefore, this pilot study also addresses prompt engineering in the context of terminology injection into LLMs with a focus on system-bound terminology. The objective was to evaluate the potential of multilingual LLM terminology work in the field of university terminology in both German and English.

Since the larger project focusses on ChatGPT only, this study aimed to assess the general suitability of LLMs for translation-oriented terminology tasks by using three different LLMs: ChatGPT (GPT-4o-mini), Perplexity ('default model') and Microsoft CoPilot. The selection of ChatGPT, Perplexity and Microsoft Copilot for this study was partly informed by their prominence in contemporary academic and professional contexts. ChatGPT was chosen due to its widespread adoption, Perplexity, as an early model integrating LLM outputs with real-time web search, is particularly suited to terminology work involving emergent terms, a phenomenon often underrepresented in static models. Microsoft Copilot, integrated into Microsoft 365, is widely used in organizations through its presence in tools like Word and Excel.

Given the practical orientation of this study, the LLMs were prompted (in German and partly in English) to address four terminology-related tasks: (1) the identification of key terms within a domain (distinct from term extraction, which typically presupposes the existence of a corpus), (2) term extraction from web-based sources (without having to compile a corpus beforehand), (3) the generation and extraction of definitions, and (4) the establishment of concept relations. These tasks simulate realistic scenarios faced by terminologists or specialized translators who require a foundational terminological database under time constraints.

The performance of the LLM also depends on the prompt being used. Selecting the right (user) prompt involves understanding user intent, model understanding and the specificity of the domain. Clear, specific prompts tailored to the task and any necessary constraints help guide the model to produce better results (Ekin, 2023). Prompt engineering principles (Bozkurt, 2024; Chen, Zhang, Langrené, & Zhu, 2013; Saleem, 2024) are a dime a dozen, ranging from general guidelines to prompt engineering frameworks. In this study, the CARE and RACE frameworks were used. CARE (context, action, result, example) consists of context, i.e. background information, action as the definition of the tasks to be completed, result to state the expected outcome and example to provide the LLM with concrete examples of what the output should be. RACE (role, action, context, expectation), on the other hand, focusses on the role, also sometimes referred to as persona the LLM should assume, action and context (which are similar to CARE) and expectation to specify the expected result. Additionally, these prompts were improved by LLMs for optimizing prompts. Moreover, prompt chaining, whereby a large task is broken into a sequence of smaller subtasks, each handled by its own prompt, was used and the domain specificity, system boundness and context dependence of terminology was considered in the prompt (Table 1).

**Table 1:** Consideration of domain specificity, system boundness and context dependence of terminology in the prompts

| Terminology characteristics | Aspect | Considered in the prompt |
|---|---|---|
| **Domain specificity** | University terminology; studies (subdomain); admission (focus) | "in the university context", "in the field of university admission" |
| **System boundness (system)** | (Austrian) university system | "Austrian higher education system", "universities in Austria" |
| **System boundness (language variety)** | Austrian | "Austrian German", "the Austrian variety of the German language" |
| **Context dependence** | University vs. university of applied sciences; terminological variation; certain universities | By including source hierarchy, domains (ac.at) from which terms and definitions should be extracted and varying the terminology used in the prompt, e.g. "university" or "higher education system"; "Benennungen" or "Termini" (in German) |

To determine whether the issue was simply finding the 'right prompt' or if it was influenced by factors beyond a single model, three different LLMs were tested. The goal was not to compare these models (or to compare them with traditional corpus analysis tools) but to gain a broader understanding of how useful LLMs are for terminology-related tasks. By using multiple models, the analysis was not limited to just one LLM, allowing for more comprehensive conclusions. This study adopts a qualitative approach to analyzing both the design of prompts and the outputs generated by the language model, with a focus on understanding LLM capabilities in specific terminological contexts, including domain-specific and system-bound (language-variety-specific) terminology. Therefore, the analysis focusses on the following aspects: 1) If the term actually exists (or is hallucinated); 2) If the term is bound to the correct system (e.g. Austrian university terminology or corporate language, if prompted) and 3) if the term is specific to the domain (and not from any other domain), i.e. the university or higher education domain (Table 2).

**Table 2:** Criteria and aspects considered in analyzing the LLM output

| Criterion | Subcriterion | Guiding questions |
|---|---|---|
| **Term existence** | Real term (vs hallucination, pseudo-terminology) | Is the term real and used in recognized sources (e.g. termbases, glossaries)? Has the LLM generated a non-existent or fabricated term? |
| **Domain specificity** | University terminology; studies (subdomain); admission (focus) | Does the term belong to the relevant specialized (sub-)field? |
| **System boundness** | Correct system | Is the term from the (Austrian or British) university system? Does the term cover the relevant language variety? |
| **Context dependence** | Corporate language (if requested) | Is the term used by the relevant university? Is it the preferred term (at the university)? |

## 3 Results

LLMs face challenges in several key areas of bilingual terminology work. The three LLMs analyzed in this study struggle with completing multiple steps or sequences, even when given step-by-step instructions within a single prompt. Additionally, they often fail to provide accurate terminological definitions, especially when requested in structured formats like tables. Moreover, they often do not provide original terms and definitions but translations of (German) terms and definitions (in English). LLMs also tend to mix Austrian university terminology with terms from the German higher education system, making it difficult to focus exclusively on the desired system, language variety and context. Lastly, when extracting definitions from websites, the models can produce inconsistent results unless the task is confined to a single, focused source.
Despite prompts that took the domain specificity, the system boundness and the context dependence of terminology into account, it was not possible to achieve the desired result. In some cases, the LLM outputs did not differentiate between the Austrian and German higher education systems. As a result, the outputs, which should have been related to the Austrian university system, contained terms from both contexts. For example, *Numerus Clausus (NC)* is not a restriction for university admission in Austria, whereas in Germany it is. The analyzed LLMs generally do not distinguish between Austrian and German university terminology despite prompts specifying the ‘Austrian university system’ or

'Austrian German'. However, they occasionally generate university-specific terms, even without being prompted, including proprietary system names like *u:space*, a platform of the University of Vienna. Also, when prompted for relations between concepts, terms from the German university system were included. While the concept relations generated by the LLMs are generally usable, they should be approached with caution. The terminologist must possess the ability to differentiate between 'German' and 'Austrian' university terminology in order to develop an accurate concept system based on the LLM output. For example, ChatGPT defined the following subordinate concepts for admission to studies (*Zulassung zum Studium*): „allgemeine Universitätsreife, fachgebundene Universitätsreife, Studienberechtigungsprüfung, Berufsreifeprüfung, Quotenregelung". While the results may seem plausible to non-experts, they contain pseudo-terminology (or hallucinations) as *fachgebundene Universitätsreife* is not a term used within the Austrian higher education system. The correct, albeit not commonly used, term would be *fachgebundene Hochschulreife*.
In the case of bilingual terminology work, in which a RACE prompt (illustrated in the Appendix) was used, the LLMs did not provide original definitions for the English or German terms provided as, in some cases, the LLMs just translated the definitions into the other language, thereby even inventing terms. For example, *conditional offer* and *unconditional offer* refer to university admissions with or without certain conditions in the English higher education system. The German terms *unbedingte Zulassung* and *bedingte Zulassung* as provided by ChatGPT are, however, not used in Austria at all. This means that the LLM outputs were often not useful to create or assess relationships between concepts and to prepare concept systems for the Austrian and (British) English higher education systems. Even by varying the specification of the respective language variety (e.g. 'Austrian German' or 'Austrian variety of German'), as well as the further specification of the system (e.g. 'Austrian higher education system', sometimes also 'at the University of Vienna') and the context, the LLM outputs could not be significantly improved.
Although the quality of the prompt has a significant impact on the quality of the LLM output, the characteristics of each LLM also play a role. The user's knowledge of these characteristics is termed 'model understanding' (Ekin, 2023). These characteristics include, for example, how up-to-date the training data are and how the training data for German are distributed across the German, Austrian, Swiss (and other) varieties of German. This is aggravated by the fact that the providers of the LLMs are often non-transparent with regard to such information. It is equally opaque whether the studied LLMs use English as a pivot language (i.e. translate the prompt into English), when the user enters German prompts, before returning the output to German. Some outputs allude to that, for example, the term *Notendurchschnitt (GPA)* was output by Microsoft CoPilot. However, the English abbreviation *GPA (grade point average)* is not a common abbreviation for *Notendurchschnitt* in German. This may lead to biases in multilingual terminology work and specialized translation in general.

## 4 Discussion

The variability of LLM outputs, even when using the same prompts, presents significant challenges in terminology-related tasks. Problems such as the creation of pseudo-terminology or inconsistencies highlight the critical importance of domain-specific expertise when using LLMs for terminology work. Without sufficient knowledge in the relevant field, such as Austrian or British university terminology, users may struggle to evaluate the quality and relevance of the generated output. This underscores the need for LLM literacy and a critical approach to LLM use in such specialized tasks. Human expertise remains crucial to meet the specific demands of terminology work, even when LLMs are involved. One key factor influencing the quality of LLM output is prompt engineering. The iterative process of refining prompts to better suit the task at hand is essential (Ekin, 2023), particularly in terminology work where domain specificity is vital. LLMs, unlike traditional tools such as corpus analysis software, require well-crafted prompts to produce accurate and relevant terminology outputs. In contrast, traditional tools can provide frequent terms within a domain without requiring extensive domain-specific input.

Despite their usefulness, LLMs do not necessarily enhance the productivity of terminologists. This study suggests that applying prompt chaining (breaking down tasks into smaller, sequential prompts) yields better results than attempting to address everything in a single prompt. This method is particularly important in multilingual terminology work, where equivalences between languages need to be established. However, LLMs as tools in terminology work can also be time-consuming, requiring terminologists to carefully scrutinize the output, verify sources and ensure the accuracy of definitions and web sources provided by the model. LLMs pose several challenges for terminology work, including biases and multilingual limitations. Their reliance on predominantly English training data (Wang et al., 2024) affects multilingual terminology since terms may be inherently altered by being filtered through English (Heinisch, in print), which makes it difficult to find equivalents in languages, other than English. Biases in, or a lack of training data can skew terminology work, particularly in emerging or niche domains (Heinisch, in print), and LLMs' tendency to hallucinate (terms) further complicates their reliability. Moreover, ecological concerns related to the energy consumption and carbon footprint of LLMs (Rojas, 2024) should be considered when choosing appropriate tools for terminology tasks. The limitations of this study lie in the selection of the LLMs and the prompting frameworks used: the analyzed models are not representative of all commercially available LLMs. Since terminology work is often multilingual (as demonstrated in this study), future research could include models with a stronger multilingual focus, such as EuroLLM (Martins et al., 2025). Furthermore, future research could employ more advanced prompting strategies and frameworks, as those used in this study were intentionally kept simple and concise. Moreover, the larger UniTermGPT project intends to include more language varieties as well as additional annotators. Given the study's focus on the practical application of LLMs by terminologists and specialized translators, no German fine-tuned models were used. The aim was to reflect realistic workflows using a single general-purpose model, particularly for high-resource languages like German and English, which are typically well-supported by such models. While Retrieval-Augmented Generation (RAG) enables LLMs to access supplementary information, Terminology-Augmented Generation (TAG) (Fleischmann & Lang, 2025) represents a complementary approach tailored to domain-specific language use. TAG integrates several components: deterministic retrieval from structured terminological databases (as opposed to probabilistic retrieval from vectorized data), the generation of precise and processable outputs in standardized or prose-like terminology formats and real-time access to terminology resources via APIs (Fleischmann & Lang, 2025). This methodology is particularly beneficial for content generation tasks that require adherence to domain-specific or corporate language norms, such as specialized translation or technical communication, where consistent use of (validated) terminology is essential.

## 5 Conclusion

While large language models offer promising possibilities for terminology work, it is evident that not all domains, languages and language varieties are equally supported by these systems. The presence of biases and hallucinations poses significant challenges in multilingual and domain-specific terminology tasks, underscoring the importance of human expertise in mitigating these issues. To ensure reliable outputs, LLM literacy is essential for users engaging with these tools in terminology work. Furthermore, prompt engineering plays a crucial role in shaping the quality of the LLM's responses, though the inherent characteristics of the model, such as its training data and its capabilities, also influence the results. In some cases, smaller, fine-tuned models focused on terminology tasks or even traditional tools like corpus analysis tools may be more suitable. The balance between LLM usage and traditional terminology tools, as well as the expertise required to navigate LLM outputs, is crucial for effective and high-quality terminology work.

## Acknowledgement

This paper was funded by the EC-MCSA Seal of Excellence Programme of the Autonomous Province of Bozen/Bolzano – Department for Innovation, Research and University, project *University terminology in German in the age of ChatGPT (UniTermGPT).*

## References


Bozkurt, A. (2024). Tell Me Your Prompts and I Will Make Them True: The Alchemy of Prompt Engineering and Generative AI. *Open Praxis, 16*(2), 111–118. Retrieved from https://search.informit.org/doi/pdf/10.3316/informit.T2024041000014390073541090

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2013). *Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review.* Retrieved from http://arxiv.org/pdf/2310.14735

Drewer, P., & Schmitz, K.-D. (2017). *Terminologiemanagement: Grundlagen - Methoden - Werkzeuge.* Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-662-53315-4

Ekin, S. (2023). *Prompt Engineering for ChatGPT: A Quick Guide to Techniques, Tips, and Best Practices.* Institute of Electrical and Electronics Engineers (IEEE). doi: 10.36227/techrxiv.22683919.v1

Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models.* Retrieved from https://doi.org/10.48550/arXiv.2303.10130

Fleischmann, K., & Lang, C. (2025). Terminologie in der KI. Wie mit Terminologie der Output von LLMs und GenAI optimiert werden kann. In P. Drewer, F. Mayer, & D. Pulitano (Eds.), *Terminologie in der KI – KI in der Terminologie. Akten des Symposions Worms, 27.–29. März 2025* (pp. 83–95). München / Karlsruhe / Bern: Deutscher Terminologie-Tag e.V.

Hamm, J. (2025). Terminologische Konsistenz und generative KI – ein Perfect Match? Produktiver Einsatz von Sprachmodellen im Terminologiemanagement und beim Post-Editing. In P. Drewer, F. Mayer, & D. Pulitano (Eds.), *Terminologie in der KI – KI in der Terminologie. Akten des Symposions Worms, 27.–29. März 2025* (pp. 151–163). München / Karlsruhe / Bern: Deutscher Terminologie-Tag e.V.

Heinisch, B. (2020). Sprachvarietätenabhängige Terminologie in der neuronalen maschinellen Übersetzung: Eine Analyse in der Sprachrichtung Englisch-Deutsch mit Schwerpunkt auf der österreichischen Varietät der deutschen Sprache. In C. Schöch (Ed.), *DHd 2020 Spielräume: Digital Humanities zwischen Modellierung und Interpretation. Konferenzabstracts* (pp. 211–214). doi: 10.5281/zenodo.4621962

Heinisch, B. (in print). Next-Gen Terminology: Transforming Terminology Work with Large Language Models. *Across Languages and Cultures*.

Hohnhold, I. (1990). *Übersetzungsorientierte Terminologiearbeit: Eine Grundlegung für Praktiker.* Stuttgart: InTra, 1. Fachübersetzergenossenschaft. Retrieved from http://media.obvsg.at/AC00246114-1001

*ISO 1087:2019 Terminology work and terminology science — Vocabulary* (2nd ed.). (2019). https://www.iso.org/standard/73906.html. (Published by International Organization for Standardization (ISO))

*ISO 704:2022 Terminology work — Principles and methods* (4th ed.). (2022). https://

www.iso.org/standard/81503.html. (Published by International Organization for Standardization (ISO))

Martins, P. H., Fernandes, P., Alves, J., Guerreiro, N. M., Rei, R., Alves, D. M., ... Martins, A. F. (2025). EuroLLM: Multilingual Language Models for Europe. *Procedia Computer Science*, *255*, 53-62. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050925006210 (Proceedings of the Second EuroHPC user day) doi: https://doi.org/10.1016/j.procs.2025.02.260

Massion, F. (2024). Terminology in the Age of AI: The Transformation of Terminology Theory and Practice. *Journal of Translation Studies*, *4*(1). Retrieved from https://www.peterlang.com/document/1495905 doi: 10.3726/JTS012024.04

Reineke, D. (2023). Terminologiearbeit mit ChatGPT & Co. *Fachzeitschrift für Terminologie*, *19*(1), 25–28. Retrieved from http://dttev.org/images/edition/ausgaben/edition-2023-1-e-version.pdf

Rojas, S. (2024). Evaluating the Environmental Impact of Large Language Models: Sustainable Approaches and Practices. *Innovative Computer Sciences Journal*, *10*(1), 1–6. Retrieved from https://innovatesci-publishers.com/index.php/ICSJ/article/view/153

Rothwell, A., Moorkens, J., Fernández-Parra, M., Drugan, J., & Austermühl, F. (2023). *Translation tools and technologies* (1st edition ed.). London New York: Routledge, Taylor & Francis Group. doi: 10.4324/9781003160793

Saleem, M. (2024). *11 ChatGPT Prompt Frameworks Every Marketer Should Know.* Retrieved from https://buttercms.com/blog/chatgpt-prompt-frameworks/

Siu, S. C. (2023). ChatGPT and GPT-4 for Professional Translators: Exploring the Potential of Large Language Models in Translation. *SSRN Electronic Journal*. doi: 10.2139/ssrn.4448091

Steurs, F., de Wachter, K., & de Malsche, E. (2015). Terminology tools. In H. J. Kockaert & F. Steurs (Eds.), *Handbook of terminology* (pp. 222–249). Amsterdam: John Benjamins Publishing Company.

Wang, W., Tu, Z., Chen, C., Yuan, Y., Huang, J.-t., Jiao, W., & Lyu, M. (2024). All Languages Matter: On the Multilingual Safety of LLMs. In *Findings of the Association for Computational Linguistics ACL 2024* (pp. 5865–5877). Stroudsburg, PA, USA: Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.349

Wüster, E. (1974). Die Allgemeine Terminologielehre - Ein Grenzgebiet zwischen Sprachwissenschaft, Logik, Ontologie, Informatik und den Sachwissenschaften. *Linguistics*, *12*(119), 61–106. Retrieved 2025-05-21, from https://doi.org/10.1515/ling.1974.12.119.61 doi: doi:10.1515/ling.1974.12.119.61

## Appendix

The appendix contains examples of prompts (according to the RACE and CARE frameworks) used in this study. The LLMs were prompted in German or English depending on the task at hand: For example, if the task was related to Austrian university terminology, the prompt was written in German, and if the task was solely related to the British university terminology, the prompt was in English. As all the German prompts will be made available as part of the larger UniTermGPT project, the following section only gives examples of the prompts in English (including translations of the German prompts). First, the different aspects of the selected prompting frameworks are illustrated. Second, examples of full prompts are provided. Third, an example of a prompt improved by *Prompt Maker* by Ruben Hassid is shown.

### Selected prompting frameworks applied to terminology tasks

The two selected prompting frameworks (RACE and CARE) applied to terminology tasks:

**Table 3:** Prompting frameworks applied to terminology tasks

| Task | CARE prompt | RACE prompt |
|---|---|---|
| **(1) Term identification** | **Context:** Identify university admission terminology in Austria (and the UK).<br>**Action:** Create list of key terms.<br>**Result:** A list of admission-related terms (with brief descriptions).<br>**Example:** transcript of records = official document summarizing a university student's academic performance and progress to date. | **Role:** Terminologist for higher education.<br>**Action:** Generate list of admission terms.<br>**Context:** Austrian and British university system.<br>**Expectation:** 10–15 terms per system (with brief definitions). |

| Task | CARE prompt | RACE prompt |
|---|---|---|
| **(2) Term extraction** | **Context:** Analyze university admission terminology in Austria (and the UK).<br>**Action:** Extract relevant terms from websites or authoritative sources.<br>**Result:** A glossary of admission-related terms (with source).<br>**Example:** Sammelzeugnis = Zeugnis über alle absolvierten Prüfungen eines Studierenden an einer Universität. | **Role:** Terminologist for higher education.<br>**Action:** Generate list of admission terms from university websites.<br>**Context:** Austrian and British university system.<br>**Expectation:** 10–15 terms per system (with source). |
| **(3) Definition generation or extraction** | **Context:** List of terms collected from domain of university admission.<br>**Action:** Define each term using ISO 704/1087 principles.<br>**Result:** Structured definitions in German/English.<br>**Example:** "Zulassungsbescheid" – Verwaltungsakt, mit dem eine Universität Studienwerber*innen formal mitteilt, dass sie für ein Studium in einer bestimmten Studienrichtung unter den angegebenen Bedingungen aufgenommen sind. | **Role:** Expert in ISO-compliant terminology work.<br>**Action:** Write precise definitions.<br>**Context:** Multilingual terminology work and management.<br>**Expectation:** 10-15 definitions of concepts in German (and English). |
| **(4) Concept relations** | **Context:** In terminology, concepts are related to each other.<br>**Action:** Identify hierarchical (generic and partitive) (and associative) relations.<br>**Result:** Concept system or structure with relation types.<br>**Example:** "University" → "Faculty" (partitive). | **Role:** Terminologist comparing university systems.<br>**Action:** Map concept relations.<br>**Context:** Prepare concept system in German / English.<br>**Expectation:** Hierarchical or associative concept relations. |

### Examples of full prompts according to the selected prompting frameworks

Examples of full prompts (mainly translated from German) according to the RACE and CARE prompting frameworks for the four selected terminology tasks are listed in the following:

**Table 4:** Prompting frameworks for terminology tasks using prose-style prompts

| Task | CARE prompt | RACE prompt |
|---|---|---|
| (1) Term identification | In the context of university admission in Austria and the UK, generate a list of the most frequently used terms in this domain. Your task is to identify and list core terminology used in university admission processes. The result should be a bilingual list (Austrian German–British English) of 10 terms including definitions according to terminological principles. For example: Studienwerber*in: Person, die an einer Universität die Zulassung zu einem bestimmten Studium beantragt. – Applicant: person who has submitted an application for admission to a university. | As a terminologist specialising in higher education, identify key terms related to university admissions in both Austria and the UK. Consider how admission is structured in each country. Your output should be a list of 10 admission-related terms per system, each with a short definition in context. |

| Task | CARE prompt | RACE prompt |
|---|---|---|
| (2) Term extraction | For terminology management in the Austrian university system, using official Austrian websites or normative documents, extract key terms related to the study admission process in Austria. This includes identifying domain-specific terminology from legal texts or university materials. Provide the term, its source (and a short contextual definition) in a three-column table. For example: 1) Studieneingangs- und Orientierungsphase, 2) Angebot von Lehrveranstaltungen aus den das jeweilige Diplom- oder Bachelorstudium besonders kennzeichnenden Fächern, das der Information und der Orientierung der Studienanfängerinnen und Studienanfänger dient, 3) `URL`. | Act as a terminology expert conducting web-based term extraction in the field of Austrian university terminology. Focus on extracting university admission terminology from Austrian institutional websites (e.g. ministry or university portals). Present 10–15 relevant terms from the Austrian higher education system and include their source (URL or document title). |

| Task | CARE prompt | RACE prompt |
|---|---|---|
| (3) Definition generation or extraction | For comparative terminology work in the university sector in German and English, compile a list of the most common terms in the field of university admissions. Add definitions for these terms from Austrian and British websites and preferably from normative or official sources, such as laws or documents from authorities or organisations. Compare terms from the Austrian and British higher education systems. Here is an example: Sammelzeugnis = 'Zeugnis über alle absolvierten Prüfungen eines Studierenden an einer Universität', the English equivalent is: transcript of records = 'official document summarising a university student's academic performance and progress to date'. Add the URL of the website where you found the definitions. | You are a terminologist and need definitions of terms in the university context in Austrian German and British English in the field of university admissions. Given a preselected list of terms related to university admission, define each term using principles from ISO 704 and ISO 1087. Where possible, extract official definitions from trusted sources (e.g. laws or university guidelines). Create a table with 6 columns: 1st column: German term. 2nd column: German definition for this term. 3rd column: Source of the definition from the Internet by specifying a real website from Austria (domain: `ac.at`). 4th column: English term. 5th column: English definition. 6th column: Source of the definition from the Internet by specifying a real website from Great Britain (domain: `ac.uk`). |
| (4) Concept relations | In the context of terminological concept systems, analyze the relationships between the concepts (and terms) identified in the Austrian university context. Your task is to construct a concept structure for the field of university admission in Austria showing hierarchical (generic and partitive) or associative relationships. The result should support the creation of (comparative) concept systems in multilingual terminology work, e.g. University → Faculty (partitive). | As an ISO-compliant terminologist comparing the Austrian and British higher education systems, identify concept relationships among university admission terminology. Develop a concept system that includes hierarchy (e.g. admission document → Zulassungsbescheid) and differences between the Austrian and British university admission systems. Document the structure clearly for bilingual comparison. |

#### Example of prompt improved by Prompt Maker LLM

Example of prompt (in German) improved by *Prompt Maker* by Ruben Hassid (although the terminology of terminology is not correct in the improved prompt, e.g. *Fachbegriff* instead of *Benennung*, the LLM output with the improved prompt was not changed, except for the headings in the sample table):

**Promptbeschreibung: Terminologiearbeit im Hochschulbereich (AT/UK)**
**Rolle & Kontext:** Agiere als Experte für vergleichende Terminologiearbeit im Hochschulbereich mit Spezialisierung auf Studienzulassungsverfahren in Österreich und Großbritannien. Deine Aufgabe ist es, eine präzise, systematische Liste der häufigsten Fachbegriffe in diesem Bereich zu erstellen und die entsprechenden Begriffe auf Deutsch und Englisch zu vergleichen.
**Ziel & Anforderungen:**

1. Erstelle eine umfassende Liste relevanter Begriffe aus dem Bereich der Studienzulassung.
2. Für jeden Begriff liefere folgende Informationen:
   - **Deutsche Bezeichnung**
   - **Englische Entsprechung (britisches Hochschulsystem)**
   - **Definition in beiden Sprachen**, basierend auf offiziellen Quellen (z. B. gesetzliche Regelungen, Dokumente von Hochschulen oder Behörden)
   - **Quelle der Definition** (falls verfügbar)
3. Beachte spezifische Unterschiede zwischen dem österreichischen und britischen Hochschulsystem. Falls es keine exakte Entsprechung gibt, erläutere die Unterschiede.

**Beispieltabelle**

**Table 5:** Vergleich von Fachbegriffen der Studienzulassung (AT/UK)

| **Deutsche Benennung** | **Englische Benennung** | **Deutsche Definition** | **Quelle Def. DE** | **Englische Definition** | **Quelle Def. EN** |
|---|---|---|---|---|---|
| Sammel-zeugnis | transcript of records | Zeugnis über alle absolvierten Prüfungen eines Studierenden an einer Universität. | [Name der Quelle] | Official document summarizing a university student's academic performance and progress to date. | [Name der Quelle] |

**Detaillierter Arbeitsablauf**

1. **Recherche:** Identifiziere die wichtigsten Begriffe im Bereich der Studienzulassung in Österreich und Großbritannien.

2. **Kategorisierung:** Ordne die Begriffe in sinnvolle Gruppen ein (z. B. Bewerbungsunterlagen, Zulassungsbedingungen, Prüfungsanforderungen).

3. **Definitionserstellung:** Nutze vertrauenswürdige Quellen, um exakte Definitionen bereitzustellen. Falls es keine direkte Entsprechung gibt, erkläre die Unterschiede.

4. **Formatierung:** Präsentiere die Ergebnisse in einer strukturierten Tabelle für bessere Übersichtlichkeit.

**Wichtige Hinweise:**

- Bevorzuge offizielle Quellen wie Hochschulgesetze, Webseiten von Universitäten oder Ministerien.

- Falls Begriffe Mehrdeutigkeiten aufweisen, gib eine kurze Kontext-Erklärung dazu.

- Falls es im britischen System kein direktes Äquivalent gibt, erläutere den Unterschied.

*Nimm dir Zeit und arbeite die Aufgabe **Schritt für Schritt** durch.*

**Correspondence**

Barbara Heinisch

Eurac Research
Institute for Applied Linguistics
Bolzano, Italy
barbara.heinisch@eurac.edu

Elena Volkanovska

# A Study of Errors in the Output of Large Language Models for Domain-Specific Few-Shot Named Entity Recognition


## Abstract

This paper proposes an error classification framework for a comprehensive analysis of the output that large language models (LLMs) generate in a few-shot named entity recognition (NER) task in a specialised domain. The framework should be seen as an exploratory analysis complementary to established performance metrics for NER classifiers, such as F1 score, as it accounts for outcomes possible in a few-shot, LLM-based NER task. By categorising and assessing incorrect named entity predictions quantitatively, the paper shows how the proposed error classification could support a deeper cross-model and cross-prompt performance comparison, alongside a roadmap for a guided qualitative error analysis.


## 1 Introduction

The advent of generative large language models (LLMs) created an increased interest in experimenting with few-shot methods for named entity recognition (NER). With LLMs, NER can be defined as a question-answering task, where a model is prompted to identify[1] named entities based on a named entity definition and named entity examples provided in the prompt. In real-world scenarios, the need for few-shot NER is driven by scarcity of resources, legal constraints for sharing annotated data, and the cost of annotation (Moscato, Postiglione, & Sperlí, 2023). However, the success of few-shot NER techniques is not consistent. Some studies using known NER datasets and LLMs have reported promising results (Ashok & Lipton, 2023; Epure & Hennequin, 2022; Wang et al., 2023).[2] At the same time, experiments using more specialised NER datasets, such as the one described in Section 3, do not achieve the same degree of success. Moscato et al. (2023) also mention that the success of few-shot NER in real-world deployment scenarios is yet to be proven.

This study investigates the possible causes of such inconsistencies by analysing LLMs' output in experiments that yielded F1 scores that were substantially below the task baseline. Rather than discarding the output as noise, the paper aims to identify what

[1] An effort was made to refrain from using anthropomorphising terms when describing LLMs (see Inie, Druga, Zukerman, and Bender (2024) for more information on this topic); nevertheless, this type of language is common in the context of generative language models and, in some cases, difficult to evade.

[2] Some authors acknowledge that data contamination i.e. the likelihood of the used LLMs having been previously exposed to the NER datasets might affect the outcome.

lessons can be learned by proposing a draft framework for a descriptive error analysis. To do so, the study first reviews existing approaches to error analysis in few-shot NER in Section 2, followed by a brief description of the experiments underpinning the analysed data in Section 3. The proposed error classification and the insights it provides into LLM performance are discussed in Sections 4 and 5 respectively.

## 2 Related Work

Generative pre-trained language models employed in some studies exploring few-shot methods for in- and cross-domain NER include the Pretrained Conditional Generation Model of Flan-T5-XXL (11B) (Chung et al., 2024), GPT-3.5 (Brown et al., 2020), and GPT4 (Achiam et al., 2023), all of which have been used in the study by Ashok and Lipton (2023); GPT-3 (davinci-003) used by Wang et al. (2023), and a medium-sized GPT-2 model used in few-shot NER experiments by Epure and Hennequin (2022).

These studies showed that the named entities (NEs) identified by LLMs can lead to valuable insights. Ashok and Lipton (2023) conduct a human survey of errors, where they (1) create a list containing 20 randomly selected examples of predicted named entity instances, (2) create a ground truth list containing NEs from the same sentences used to create list (1), and (3) ask three different human annotators to evaluate each entity of lists (1) and (2). The human annotators are given a definition of the NER problem relevant to the dataset from which the lists are created. The results from this evaluation show that many of the predictions could be acceptable NE candidates and were not considered errors by the human annotators.

The evaluation approach adopted by Epure and Hennequin (2022) for NER in a few-shot setting is case-insensitive and accommodates for output where the model generates an NE with a different spelling or when it fails to follow the instruction for sentences containing no entities. The study dubs as *confusion patterns* cases when the LM fails to generate the correct entity type, conflating, for example, *corporation* or *group* with *location*. The study's authors provide a brief overview of NE categories that perform well and categories that do not. Wang et al. (2023) also find that the LLM conflates *location* and *geographical entities* in a nested NER scenario.

While it is evident that language models' output is manually inspected, with researchers working in few-shot NER performing an error analysis in order to compare the effects of various prompt designs and task requirements, the insights that come from the manual inspection are mostly captured in the recommendations for prompt design in future studies. In other words, such analyses have not amounted to a systematic classification of errors identified in models' output.

**Contribution** This paper proposes a descriptive error analysis method for LLM output in a few-shot NER task on two domain-specific NER datasets. It combines categories from existing NER evaluation metrics, such as F1 scores, and error analyses encountered in previous studies on few-shot NER into a single error classification framework for model output analysis. This framework could be used to (1) gauge weak points in the task design and in the LLMs' performance and (2) make informed decisions for qualitative error analysis and iterative changes to the prompt design.

## 3 Data

**LLMs and datasets** The data analysed in this study is the LLM output from a series of few-shot NER experiments, where 7762 prompts are run on four LLMs: OpenAI's gpt-4o-2024-05-13 and gpt-4o-mini (hereinafter: gpt-4o and gpt-4o-mini), and Meta's Meta-Llama-3.1-70B-Instruct and Meta-Llama-3.1-405B-Instruct (hereinafter: Llama-70B and Llama-405B). The experiments are conducted on the test data splits of two NER datasets comprising scientific texts: Climate-Change-NER (Bhattacharjee et al., 2024) with 13 climate-change-relevant NE categories (*climate-assets*, *climate-datasets*, *climate-greenhouse-gases*, *climate-hazards*, *climate-impacts*, *climate-mitigations*, *climate-models*, *climate-nature*, *climate-observations*, *climate-organisms*, *climate-organizations*, *climate-problem-origins*, and *climate-properties*), and BiodivNER (Abdelmageed et al., 2022) with 6 biodiversity-relevant NE categories (*organism*, *phenomena*, *matter*, *environment*, *quality*, and *location*). The LLMs' output and dataset information are available in a dedicated GitHub repository.[3]

**Prompts** The rationale behind the prompting methodology, the prompt design, and the results for each prompt and language model are described in detail in Volkanovska (2025). The prompt design was inspired by the study of Ashok and Lipton (2023), with the final promts differing in three major ways: (1) the input/output requirement (either a Python string or a tokenized sentence i.e. a Python list of word-based tokens and their indices), (2) the number of NE categories tested, and (3) the method of selecting task examples (TEs) in the prompt. Under (1), the prompts can have either *string-based* or *token-based* input (TEs) and output (a requirement for the model to generate an answer in a format that corresponds to the TEs). Under (2), there are *full prompts*, where models are tested on the complete set of NE categories, and *cluster prompts*, where the models are tested on subgroups of NE categories.

The category *full prompts* contains 6 prompt versions, which differ in the number of TEs provided to the model (3, 4 or 5). Regarding *cluster prompts*, named entities are divided into clusters of categories. For Climate-Change-NER, the clusters are: (1) *climate-hazards*, *climate-problem-origins*, *climate-greenhouse-gases*; (2) *climate-impacts*, *climate-assets*, *climate-nature*, *climate-organisms*; (3) *climate-datasets*, *climate-models*, *climate-observations*, *climate-properties*, and (4) *climate-mitigations*, *climate-organisations*. For BiodivNER, the three clusters are: (1) *environment*, *location*; (2) *organism*, *matter*, and (3) *phenomena*, *quality*. Finally, under (3), TEs contained either randomly selected sentences from the train data split, or sentences with a high semantic similarity score to the sentence the model was to annotate. Semantic similarity scores were calculated with the library sentence-transformers (Reimers & Gurevych, 2019) and the model *sentence-transformers/stsbdistilroberta-base-v2*.

[3] https://github.com/volkanovska/NER-annotation-with-LLMs

The different prompting scenarios showed that token-based prompts performed, on average, slightly better than string-based prompts. For the former, LLMs' averaged F1 scores[4] ranged between 0.27 (lowest) and 0.41 (highest). For string-based prompts, the averaged F1 scores ranged from 0.28 to 0.39. LLMs generally performed better when there were more TEs, while the TEs' similarity to the task sentence had a greater impact on the result when the original dataset contained some noise, most likely introduced by text extraction from PDF sources. As token-based prompts performed slightly better than string-based prompts, the error analysis proposed in this paper is conducted on the output from token-based prompts. See Appendix 7 for a prompt example.

## 4 Methodology

In the context of this study, *error* encompasses all instances where the model's output does not fully match the correct answer. For a candidate entity to be considered *correct*, there must be a full span-and-category match between the candidate and the gold standard named entity. Partial matches, as well as minor hallucinations, such as an incorrectly spelled entity type, are considered errors.

The LLM output of named entity candidates is thus analysed as follows: first, a count of all predicted entities is provided. Perfect and missed matches of (entity, entity category) are counted by comparing the model's predictions to the gold standard. Then, predicted entities that are not **perfect** matches are divided into four error classes: (1) LLM output where a valid NE instance[5] is assigned the wrong category from the set of valid NE categories[6] (dubbed **sources of confusion**), (2) a valid NE category is assigned to spans that have not been identified as named entities in the original dataset (**possible candidates**), (3) a valid named entity is assigned a named entity category that is not part of the original dataset (**new categories**) and (4) neither the named entity span nor the assigned entity category is valid (**pure noise**).

This error classification is a descriptive overview of the errors found in the models' output and aims to complement established evaluation metrics. Missed and perfect matches, as well as *sources of confusion* and *possible candidates*, are output categories that have been accounted for in existing evaluation metrics.[7] The classes *new categories* and *pure noise* are added to capture LLM-specific issues arising from LLMs' "hallucinations".

**Counting error instances** For *cluster prompts*, the counts of each error class represent the number of unique error instances found in each error class per cluster. For example, in cluster 1 of Climate-Change-NER (*climate-hazards*, *climate-problem-origins*, *climate-greenhouse-gases*), errors of the class **sources of confusion** are counted for this cluster only for each LLM. For *full prompts*, the reported counts per error class

[4]An average of the F1 scores calculated for each prompt.

[5]*Valid NE instance* is an instance that exists as a named entity span in the dataset.

[6]*Valid NE category* is a named entity category that is part of the dataset's entity types.

[7]These include missed entity spans, hypothesised entity spans where there are none, entity spans that are assigned the wrong category, entity spans with incorrect boundaries and correct NE category, and entity spans with incorrect boundaries and incorrect NE category.

represent the average from the six full-prompt versions. For example, the reported count of the error class **sources of confusion** will be the sum of the error counts for each of the six prompt versions[8] divided by six. The Python script for classification of error instances, and the tables with error counts for each error class and each model are available in the GitHub repository.

**Points of comparison** In a supervised NE recognition task, a model's output is only compared to the test split of the gold dataset, given that the train and development splits are used in the model's training. In the few-shot scenario described in Section 3, however, the model had not been exposed to the development set at all and had been exposed to a maximum of five sentences from the train set. For this reason, the LLMs' output is also compared to the combinations *test and train* and *test and development* data splits of the gold standard dataset. Differences in the number of missed matches between a model's predictions and the gold standard across the three points of comparison will show whether some of the candidates generated by the model are valid entities in the development and the train data splits.

In terms of F1-score, comparative performance has been seen between the larger models, gpt-4o and LLama-405B, and the smaller models, gpt-4o-mini and Llama-70B. For this reason, error classes are further analysed per two groups of models: **large** and **small**. The error class ranking for individual models is available in the GitHub repository.

## 5 Results

Tables 1 and 2 summarize the error class counts per each prompt type and model, shown as percentages: the **missed** column shows the percentage of missed unique gold entities, while the other four columns show the percentage the respective error class has in the total number of unique predicted entity candidates. The columns **predicted** and **gold** capture the unique pairs of (named entity, named entity type) in a model's output and in the gold dataset, respectively. The recurrence of instances is not taken into account for the calculation of percentages in the two tables, as the focus is on the portion of unique instances in each error class; however, repeated occurrences are accounted for in the rankings of most-frequently represented categories and named entities in each error class; see the discussion under *Zeroing in on error classes* for more details.

All models generate a substantially higher number of entity candidates in a cluster-prompt scenario in Climate-Change-NER and across all prompt scenarios in BiodivNER. In terms of model families, Llama models generate, on average, more entity candidates, while OpenAI models tend to be more conservative.

A higher number of entity candidates does not necessarily translate into better performance, as can be seen from the error count results for smaller models, which

[8] Prompts with random task examples with 3, 4 and 5 shots, and prompts with similar task examples with 3, 4 and 5 shots.

| Model | Prompt type | Predicted | Gold | Missed (% of gold) | | | Sources of confusion (% of predicted) | | | Possible candidates (% of predicted) | | | New categories (% of predicted) | | | Pure noise (% of predicted) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev |
| Llama-70B | cluster prompts 1 | 197 | 54 | 0,37 | 0,33 | 0,37 | 0,01 | 0,02 | 0,01 | 0,81 | 0,79 | 0,81 | 0,01 | 0,01 | 0,01 | 0,01 | 0,01 | 0,01 |
| Llama-70B | cluster prompts 2 | 341 | 131 | 0,50 | 0,43 | 0,44 | 0,02 | 0,04 | 0,03 | 0,78 | 0,74 | 0,76 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-70B | cluster prompts 3 | 329 | 177 | 0,50 | 0,47 | 0,50 | 0,03 | 0,03 | 0,03 | 0,70 | 0,68 | 0,70 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-70B | cluster prompts 4 | 236 | 61 | 0,52 | 0,52 | 0,51 | 0,00 | 0,00 | 0,00 | 0,87 | 0,87 | 0,86 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-70B | full prompts (avg.) | 440 | 423 | 0,67 | 0,64 | 0,66 | 0,11 | 0,12 | 0,11 | 0,56 | 0,53 | 0,55 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | cluster prompts 1 | 120 | 54 | 0,37 | 0,33 | 0,37 | 0,01 | 0,01 | 0,01 | 0,70 | 0,68 | 0,70 | 0,00 | 0,00 | 0,00 | 0,01 | 0,01 | 0,01 |
| Llama-405B | cluster prompts 2 | 379 | 131 | 0,56 | 0,47 | 0,53 | 0,03 | 0,03 | 0,00 | 0,82 | 0,78 | 0,81 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | cluster prompts 3 | 288 | 177 | 0,51 | 0,49 | 0,51 | 0,03 | 0,03 | 0,00 | 0,67 | 0,65 | 0,66 | 0,00 | 0,00 | 0,00 | 0,01 | 0,01 | 0,01 |
| Llama-405B | cluster prompts 4 | 97 | 61 | 0,51 | 0,51 | 0,51 | 0,01 | 0,01 | 0,01 | 0,68 | 0,68 | 0,68 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | full prompts (avg.) | 557 | 423 | 0,55 | 0,52 | 0,55 | 0,10 | 0,11 | 0,00 | 0,56 | 0,53 | 0,55 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 1 | 137 | 54 | 0,43 | 0,37 | 0,43 | 0,04 | 0,04 | 0,04 | 0,74 | 0,72 | 0,74 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 2 | 330 | 131 | 0,57 | 0,47 | 0,54 | 0,02 | 0,03 | 0,03 | 0,81 | 0,76 | 0,79 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 3 | 323 | 177 | 0,53 | 0,51 | 0,53 | 0,05 | 0,06 | 0,05 | 0,70 | 0,68 | 0,69 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 4 | 98 | 61 | 0,69 | 0,69 | 0,67 | 0,01 | 0,01 | 0,01 | 0,79 | 0,79 | 0,78 | 0,00 | 0,00 | 0,00 | 0,01 | 0,00 | 0,01 |
| gpt-4o-mini | full prompts (avg.) | 568 | 423 | 0,61 | 0,57 | 0,60 | 0,13 | 0,13 | 0,13 | 0,58 | 0,55 | 0,57 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 1 | 111 | 54 | 0,33 | 0,30 | 0,33 | 0,01 | 0,01 | 0,01 | 0,67 | 0,65 | 0,67 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 2 | 232 | 131 | 0,51 | 0,37 | 0,46 | 0,00 | 0,00 | 0,00 | 0,69 | 0,60 | 0,66 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 3 | 214 | 177 | 0,55 | 0,54 | 0,55 | 0,00 | 0,00 | 0,00 | 0,57 | 0,56 | 0,57 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 4 | 124 | 61 | 0,48 | 0,48 | 0,48 | 0,01 | 0,01 | 0,01 | 0,73 | 0,73 | 0,73 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | full prompts (avg.) | 447,5 | 423 | 0,61 | 0,57 | 0,60 | 0,00 | 0,00 | 0,00 | 0,52 | 0,48 | 0,51 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |

**Table 1:** Climate-Change-NER: Missed entities as % of gold entities and error class counts as % of predicted entities.

| Model | Prompt type | Predicted | Gold | Missed (% of gold) | | | Sources of confusion (% of predicted) | | | Possible candidates (% of predicted) | | | New categories (% of predicted) | | | Pure noise (% of predicted) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev | test | test+train | test+dev |
| Llama-70B | cluster prompts 1 | 275 | 98 | 0,55 | 0,39 | 0,48 | 0,01 | 0,01 | 0,01 | 0,83 | 0,77 | 0,80 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-70B | cluster prompts 2 | 309 | 160 | 0,54 | 0,42 | 0,51 | 0,02 | 0,02 | 0,02 | 0,66 | 0,60 | 0,65 | 0,00 | 0,00 | 0,00 | 0,08 | 0,08 | 0,08 |
| Llama-70B | cluster prompts 3 | 559 | 229 | 0,55 | 0,44 | 0,53 | 0,03 | 0,03 | 0,03 | 0,79 | 0,74 | 0,78 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-70B | full prompts (avg.) | 766,67 | 487 | 0,51 | 0,42 | 0,49 | 0,05 | 0,05 | 0,05 | 0,63 | 0,57 | 0,61 | 0,00 | 0,00 | 0,00 | 0,01 | 0,01 | 0,01 |
| Llama-405B | cluster prompts 1 | 372 | 98 | 0,48 | 0,28 | 0,41 | 0,01 | 0,01 | 0,01 | 0,85 | 0,80 | 0,83 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | cluster prompts 2 | 315 | 160 | 0,45 | 0,31 | 0,41 | 0,02 | 0,02 | 0,02 | 0,70 | 0,63 | 0,68 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | cluster prompts 3 | 688 | 229 | 0,52 | 0,41 | 0,51 | 0,02 | 0,02 | 0,02 | 0,82 | 0,78 | 0,82 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Llama-405B | full prompts (avg.) | 906,67 | 487 | 0,49 | 0,38 | 0,46 | 0,05 | 0,05 | 0,05 | 0,67 | 0,61 | 0,66 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 1 | 198 | 98 | 0,30 | 0,24 | 0,54 | 0,02 | 0,03 | 0,02 | 0,78 | 0,72 | 0,75 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 2 | 440 | 160 | 0,20 | 0,15 | 0,52 | 0,02 | 0,03 | 0,02 | 0,82 | 0,76 | 0,81 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | cluster prompts 3 | 657 | 229 | 0,20 | 0,15 | 0,56 | 0,01 | 0,01 | 0,01 | 0,84 | 0,79 | 0,84 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o-mini | full prompts (avg.) | 830 | 487 | 0,31 | 0,24 | 0,50 | 0,06 | 0,07 | 0,07 | 0,65 | 0,58 | 0,64 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 1 | 181 | 98 | 0,25 | 0,31 | 0,41 | 0,01 | 0,02 | 0,02 | 0,70 | 0,61 | 0,66 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 2 | 282 | 160 | 0,24 | 0,30 | 0,40 | 0,02 | 0,02 | 0,02 | 0,66 | 0,58 | 0,64 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | cluster prompts 3 | 502 | 229 | 0,20 | 0,34 | 0,42 | 0,02 | 0,02 | 0,02 | 0,73 | 0,68 | 0,72 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| gpt-4o | full prompts (avg.) | 710 | 487 | 0,35 | 0,40 | 0,48 | 0,04 | 0,05 | 0,05 | 0,62 | 0,54 | 0,59 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |

**Table 2:** BiodivNER: Missed entities as % of gold entities and error class counts as % of predicted entities.

generate more noise. Across all models and almost all prompt types, the number of possible candidates drops once the spans from the train split of the gold dataset are added to the comparison set. This means that the models generated spans that are part of the train split - albeit not under the right category. This tendency is present, to a lesser extent, in the comparison with the development set. The miscategorisation of entity instances also explains why error counts of the category **sources of confusion** slightly increase once the train data split is added to the comparison. Percentage-wise, the error classes **new categories** and **pure noise** have generally very low values across the two datasets and all models. This indicates that the models can “follow” the guidance for identifying entities belonging to certain categories only.

**Zeroing in on error classes** The top three categories of **possible** entity candidates in **Climate-Change-NER**, identified by larger and smaller LLMs alike, albeit in different order, are: *climate-models*, *climate-nature*, and *climate-properties*. Among the most frequent candidates for *climate-models* are instances such as *GCM* or *General Circulation Models*, which in the gold datasets are only sometimes annotated as *climate-models*, usually when the term is more narrowly defined.[9] This echoes some of the findings by Epure and Hennequin (2022), who notice that in few-shot settings, pre-trained models tend to prioritize named entity cues more than context cues. The fact that the acronym *GCM* appears both as an entity and a non-entity adds a layer of complexity in the recognition stage that the LLMs cannot resolve based on context cues i.e. the term being narrowly-defined or not. All models identify spans such as *random forests* as valid instances, which indicates that there seems to be no differentiation between a climate-specific model and a general model that can be used in a climate scenario. LLMs sometimes delete extra whitespaces found in the gold dataset. Models would thus extract *WRF-UCM* instead of *WRF - UCM* as a climate model.

In the top-three categories of the **missed** error class, the category *climate-models* came in third for large and small models alike, following *climate-nature* and *climate-properties*. It included instances of LLMs failing to extract acronyms separately from the full name of a climate model, in situations where the acronym followed the name of a climate model.[10]

Small models tend to generate more invalid categories than their larger counterparts, especially in the error class **pure noise**. The invalid NE categories range from misspellings (*climate-greenhouse-gasses*, *climate-impats*), labels that are seemingly correct but contain a combination of Latin and Cyrillic letters, to categories that are not part of the original label set at all (*climate-projects*, *climate-regulations*, *climate-study-field...*).

For **BiodivNER**, the top three categories of **possible** candidates identified by large LLMs are: *quality*, *organism*, and *environment*; a slightly different frequency ranking was noticed in smaller LLMs, namely: *organism*, *quality*, and *phenomena*. While some of the candidates could be considered valid instances, such as *guinea pig* and *termites* for *organism*, other candidates include names of organisations and people, which is not in line with the NE class description.[11]

The top three entity types in the **missed** error class for large and small models are: *quality*, *matter*, and *organism*. Some of the most frequently missed instances include *species*, *tree*, and *plant*, which are found in the error class **possible candidates** as parts of longer spans.

When it comes to “hallucinations”, models that are on the smaller side tend to generate them more frequently and in greater variety. Large models did not have any errors in the *new categories* error class, and generated only 4 invalid categories in the

[9] For example, in the span *NASA / GIS GCM*, *GCM* is annotated as a climate model.

[10] In the gold dataset, acronyms are annotated as separate entities. For example, in the span *Coupled Model Intercomparison Project Phase 5 (CMIP5)*, *Coupled Model Intercomparison Project Phase 5* and *CMIP5* are two separate entities of the type *climate-models*.

[11] The class is defined as “All individual life forms such as microorganisms, plants, animals, mammals, insects, fungi, bacteria etc.”

**pure noise** error class. Smaller models, on the other hand, generated 6 new categories for existing spans and identified 56 invalid spans across more than 15 invalid categories, including combined labels such as *organism (quality)*.

## 6 Conclusion and discussion

This paper proposes a methodology for classifying errors detected in the output of LLMs following a few-shot NER task, where NER is defined as a question-answering task with a specific output requirement. The proposed error classification provides a snapshot of how LLMs fail and a systematic comparison of the output from multiple LLMs. The descriptive error counts could serve as a basis for (a) additional quantitative and (b) guided qualitative analyses. Under (a), one may explore what percentage of the errors classified as *possible candidates* are partial matches with spans from the gold dataset. Another useful information would be the average span lengths across entity instances in different error categories, and possible variations in the lengths of sentences where entities belonging to different error categories are found. This could help steer efforts under (b), which might include a hands-on comparison of sentences where repeated error instances are found.

In this study, the counts of errors in different prompt versions (random and similar task examples with 3, 4, and 5 shots) were averaged due to the limited variations in the F1 score achieved by different prompts and the primary focus being on the comparison of the four models' performance rather than prompt-specific variations. It would be beneficial to conduct error comparison per prompt output, which might show if and how each model's generation had been affected by the prompt design.

Finally, the few-shot NER task might benefit from a (self)-verification step (Li et al., 2024; Madaan et al., 2023), where either the same model or a different model "checks" the errors classified as *possible candidates* by the *annotator* model and flags up valid entity candidates. In addition, the prompt may include an instruction for the LLM to not change the input text, which might help with cases where the model removes whitespaces in the generated texts.

## 7 Acknowledgements

The research presented in this paper was conducted within the research project InsightsNet (`https://insightsnet.org/`), which is funded by the Federal Ministry of Education and Research (BMBF) under grant no. 01UG2130A. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The author would like to thank the anonymous reviewers for their insightful feedback and comments.

## References


Abdelmageed, N., Löffler, F., Feddoul, L., Algergawy, A., Samuel, S., Gaikwad, J., ... König-Ries, B. (2022). Biodivnere: Gold standard corpora for named entity recognition and relation extraction in the biodiversity domain. *Biodiversity Data Journal*, *10*.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F., ... others (2023). Gpt-4 technical report. arxiv. *arXiv preprint arXiv:2303.08774*.

Ashok, D., & Lipton, Z. C. (2023). Promptner: Prompting for named entity recognition. *ArXiv*, *abs/2305.15444*. Retrieved from https://api.semanticscholar.org/CorpusID:258887456

Bhattacharjee, B., Trivedi, A., Muraoka, M., Ramasubramanian, M., Udagawa, T., Gurung, I., ... others (2024). Indus: Effective and efficient language models for scientific applications. *arXiv preprint arXiv:2405.10725*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... others (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, *25*(70), 1–53.

Epure, E. V., & Hennequin, R. (2022, June). Probing pre-trained auto-regressive language models for named entity typing and recognition. In N. Calzolari et al. (Eds.), *Proceedings of the thirteenth language resources and evaluation conference* (pp. 1408–1417). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2022.lrec-1.151

Inie, N., Druga, S., Zukerman, P., & Bender, E. M. (2024). From" ai" to probabilistic automation: How does anthropomorphization of technical systems descriptions influence trust? In *The 2024 acm conference on fairness, accountability, and transparency* (pp. 2322–2347).

Li, Z., Xu, X., Shen, T., Xu, C., Gu, J.-C., Lai, Y., ... Ma, S. (2024). Leveraging large language models for nlg evaluation: Advances and challenges. *arXiv preprint arXiv:2401.07103*.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., ... others (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, *36*, 46534–46594.

Moscato, V., Postiglione, M., & Sperlí, G. (2023). Few-shot named entity recognition: Definition, taxonomy and research directions. *ACM Transactions on Intelligent Systems and Technology*, *14*(5), 1–46.

Reimers, N., & Gurevych, I. (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing.* Association for Computational Linguistics. Retrieved from https://arxiv.org/abs/1908.10084

Volkanovska, E. (2025, March). Large language models as annotators of named entities in climate change and biodiversity: A preliminary study. In V. Basile, C. Bosco, F. Grasso, M. O. Ibrohim, M. Skeppstedt, & M. Stede (Eds.), *Proceedings of the 1st workshop on ecology, environment, and natural language processing (nlp4ecology2025)* (pp. 24–33). Tallinn, Estonia: University of Tartu Library. Retrieved from https://aclanthology.org/2025.nlp4ecology-1.7/

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., ... Wang, G. (2023). Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Appendix A: Prompt example

The prompt included here is motivated by the prompt used by Ashok and Lipton (2023). One major difference is that in this prompt, the LLM processes a task requirement comprised of natural language and Python code, and is instructed to generate output as a Python list. The prompt in this Appendix contains three random task examples from the dataset BiodivNER.

Definition: An entity is an ORGANISM (all individual life forms such as microorganisms, plants, animals, mammals, insects, fungi, bacteria etc.), PHENOMENA (occurring natural, biological, physical or chemical processes such as decomposition, colonisation, deforestation, as well as events, such as climate change etc.), MATTER (chemical and biological compounds, and natural elements, such as carbon, sediment, sand etc.), ENVIRONMENT (natural and man-made environments organisms live in, such as groundwater, garden, aquarium, mountain etc.), QUALITY (data parameters measured or observed, phenotypes and traits, such as volume, age, structure, morphology etc.), and LOCATION (geographic location such as China, the United States etc.).
Dates, times, and adjectives are not entities.

Named entity categories, definitions, and real-world examples.

Example 1: [[0, 'Because'], [1, 'fungal'], [2, 'pathogens'], [3, 'likely'], [4, 'have'], [5, 'similar'], [6, 'abiotic'], [7, 'requirements'], [8, 'for'], [9, 'growth'], [10, 'as'], [11, 'other'], [12, 'fungi'], [13, ','], [14, 'characterizing'], [15, 'weather'], [16, 'conditions'], [17, 'favorable'], [18, 'for'], [19, 'fungi'], [20, 'also'], [21, 'may'], [22, 'be'], [23, 'used'], [24, 'to'], [25, 'predict'], [26, 'the'], [27, 'selective'], [28, 'pressures'], [29, 'imposed'], [30, 'by'], [31, 'pathogenic'], [32, 'fungi'], [33, 'on'], [34, 'plants'], [35, 'in'], [36, 'different'], [37, 'habitats'], [38, '.']]
Answer: [['growth', 'PHENOMENA', 9, 9], ['fungi', 'ORGANISM', 12, 12], ['weather conditions', 'QUALITY', 15, 16], ['fungi', 'ORGANISM', 19, 19], ['fungi', 'ORGANISM', 32, 32], ['plants', 'ORGANISM', 34, 34], ['habitats', 'ENVIRONMENT', 37, 37]]

Example 2: [[0, '--'], [1, '-'], [2, '6'], [3, 'digit'], [4, 'metal'], [5, 'tags'], [6, 'starting'], [7, 'with'], [8, '3'], [9, 'were'], [10, 'also'], [11, 'used'], [12, 'for'], [13, 'woody'], [14, 'debris'], [15, 'items'], [16, 'CSP'], [17, 'metal'], [18, 'tag'], [19, 'number'], [20, '('], [21, 'trees'], [22, ','], [23, 'woody'], [24, 'debris'], [25, ')'], [26, '('], [27, 'TagMBa'], [28, ')'], [29, ','], [30, 'dimensionless'], [31, 'TagMBa'], [32, 'CSP'], [33, 'tree'], [34, 'individuals'], [35, 'were'], [36, 'marked'], [37, 'mostly'], [38, 'with'], [39, 'metal'], [40, 'tags'], [41, 'but'], [42, 'also'], [43, 'additional'], [44, 'tags'], [45, 'were'], [46, 'used'], [47, '.']]
Answer: [['metal', 'MATTER', 4, 4], ['woody', 'ENVIRONMENT', 13, 13], ['metal', 'MATTER', 17, 17], ['trees', 'ORGANISM', 21, 21], ['woody', 'ENVIRONMENT', 23, 23], ['tree', 'ORGANISM', 33, 33], ['metal', 'MATTER', 39, 39]]

Example 3: [[0, '('], [1, 'Phenolics'], [2, ':'], [3, 'Total'], [4, 'phenolics'], [5, 'content'], [6, 'as'], [7, 'tannic'], [8, 'acid'], [9, 'equivalent'], [10, ')'], [11, 'dimensionless'], [12, 'real'], [13, 'Secondary'], [14, 'Metabolites'], [15, 'Secondary'], [16, 'metabolites'], [17, 'are'], [18, 'organic'], [19, 'compounds'], [20, 'that'], [21, 'are'], [22, 'not'], [23, 'directly'], [24, 'involved'], [25, 'in'], [26, 'the'], [27, 'normal'], [28, 'growth'], [29, ','], [30, 'development'], [31, ','], [32, 'or'], [33, 'reproduction'], [34, 'of'], [35, 'an'], [36, 'organism'], [37, '.']]
Answer: [['tannic acid', 'MATTER', 7, 8], ['Secondary Metabolites', 'MATTER', 13, 14], ['Secondary metabolites', 'MATTER', 15, 16], ['organic compounds', 'MATTER', 18, 19], ['normal growth', 'PHENOMENA', 27, 28], ['development', 'PHENOMENA', 30, 30], ['reproduction', 'PHENOMENA', 33, 33], ['organism', 'ORGANISM', 36, 36]]

Examples of the task. The LM is given a list of indices and tokens, and is expected to retrieve the NE instance, NE category, and NE indices.

In this example, the model is presented with 3 random example sentences.

Generate ONLY a Python list with a nested list of named entities from the sentence: [[0, 'The'], [1, 'primacy'], [2, 'of'], [3, 'either'], [4, 'species'], [5, 'or'], [6, 'functional'], [7, 'group'], [8, 'richness'], [9, 'effects'], [10, 'depended'], [11, 'on'], [12, 'the'], [13, 'sequence'], [14, 'of'], [15, 'testing'], [16, 'these'], [17, 'terms'], [18, ','], [19, 'indicating'], [20, 'that'], [21, 'both'], [22, 'aspects'], [23, 'of'], [24, 'richness'], [25, 'were'], [26, 'congruent'], [27, 'and'], [28, 'complementary'], [29, 'to'], [30, 'expected'], [31, 'strong'], [32, 'effects'], [33, 'of'], [34, 'legume'], [35, 'presence'], [36, 'and'], [37, 'grass'], [38, 'presence'], [39, 'on'], [40, 'plant'], [41, 'chemical'], [42, 'composition'], [43, '.']].
one-token entity: [entity, label, index of token in list, index of token of list]
multi-token entity: [entity, label, index of first token, index of last token]
DO NOT HALLUCINATE

Instructions: The LM needs to follow the output guidelines and generate a nested Python list.

**Figure 1:** Prompt example: Three randomly selected task examples (question-answer pairs) from BiodivNER's training data.

**Correspondence**

Elena Volkanovska

Technische Universität Darmstadt
Institute of Linguistics and Literary Studies
Darmstadt, Germany
elena.volkanovska@tu-darmstadt.de

Yanming Li, Meaghan Fowlie

# GPT makes a poor AMR parser


### Abstract

This paper evaluates GPT models as out-of-the-box Abstract Meaning Representation (AMR) parsers using prompt-based strategies, including 0-shot, few-shot, Chain-of-Thought (CoT), and a two-step approach in which core arguments and non-core roles are handled separately. Our results show that GPT-3.5 and GPT-4o fall well short of state-of-the-art parsers, with a maximum Smatch score of 60 using GPT-4o in a 5-shot setting. While CoT prompting provides some interpretability, it does not improve performance. We further conduct fine-grained evaluations, revealing GPT's limited ability to handle AMR-specific linguistic structures and complex semantic roles. Our findings suggest that, despite recent advances, GPT models are not yet suitable as standalone AMR parsers.


## 1 Introduction

Much of Abstract Meaning Representation (AMR) parsing is currently concentrated on fine-tuning pre-trained language models such as BART (Lewis et al., 2020). Newer Large Language Models (LLMs) such as GPT bring a new paradigm for NLP research: prompting. LLMs also show impressive "reasoning" capabilities and a certain kind of interpretability with Chain-of-Thought (CoT) prompting. With prompt-based learning, an LLM might be capable of just about any NLP task, if the right prompt and mapping from output text to task output can be found (P. Liu et al., 2023).

This paper explores the possibility that AMR parsing is possible if the requested output is in PENMAN notation. In this paper, we apply a variety of prompting strategies to induce GPT to do AMR parsing. We demonstrate that GPT models are insufficient as AMR parsers. Our work also results in two main findings. First, Chain-of-Thought prompting is also ineffective, though it offers analytical insights and shows some potential. Second, decomposing the task into identifying core argument roles and modifiers did not improve performance. Beyond these findings, we contribute an additional fine-grained evaluation for deeper analysis.

## 2 Background & Related Work

**Abstract Meaning Representation** An AMR (Figure 1) is composed of labelled nodes and edges, where nodes represent *concepts* – roughly the words or semantic units of the sentence – and edges represent the relationships between them. Formally, an AMR graph can be expressed as a set of triples $(s, r, t)$, where $s$ is the source concept (head),

$r$ is a semantic relation label (e.g. `:ARG0`, `:mod`), and $t$ is the target concept or value (Goodman, 2020).

AMR guidelines[1] specify details such as the use of PropBank (Choi, Bonial, & Palmer, 2010) verb senses (e.g. `receive-01`) and numbered arguments (e.g. `:ARG0`), named entity subgraphs, and negation, indicated with `(:polarity -)`. The numerical suffix in `receive-01` denotes a verb sense (here: `get something`), while `:ARG0` typically denotes the receiver. AMRs are written in Penman notation, a parenthesis-based representation for nested graphs, which allows text-based models to generate them directly (van Noord & Bos, 2017).

**AMR Parsing** is the task of generating an AMR given a sentence. Existing AMR parsers mainly fall into three categories: transition-based models, sequence-to-graph models, and sequence-to-sequence (seq2seq) models. Transition-based models generate new nodes, edges, or subgraphs based on the words of the sentence (Fernandez Astudillo, Ballesteros, Naseem, Blodgett, & Florian, 2020; Lindemann, Groschwitz, & Koller, 2020; Naseem et al., 2019; Peng, Gildea, & Satta, 2018; Zhou, Naseem, Fernandez Astudillo, & Florian, 2021). Sequence-to-graph models derive the graph from existing nodes without transition processes, directly extending new nodes and edges (D. Cai & Lam, 2020; Zhang, Ma, Duh, & Van Durme, 2019). Seq2seq models directly generate the text format of AMRs from raw sentences (Bai, Chen, & Zhang, 2022; Blloshmi, Tripodi, & Navigli, 2020; Lee et al., 2022; van Noord & Bos, 2017; Vasylenko, Huguet Cabot, Martínez Lorenzo, & Navigli, 2023). We use GPT as a seq2seq model.

A parallel study by Ettinger, Hwang, Pyatkin, Bhagavatula, and Choi (2023) also investigates AMR parsing with GPT models, using similar prompting strategies such as 0-shot and 5-shot prompting. While our results are slightly better, both studies remain far from state-of-the-art performance. Compared to their work, our experiments are conducted on larger datasets, include novel prompting strategies, and provide fine-grained analysis using GrAPES (Groschwitz, Cohen, Donatelli, & Fowlie, 2023). We also systematically evaluate the GPT model's ability to generate well-formed AMRs (termed Parsability), and show that post-processing significantly improves Parsability to over 90% in all settings except 0-shot.

**In-context learning/$k$-shot/few-shot prompting** is a gradient-free "learning" strategy for language models that provides $k$ task-related example question-answer pairs before asking the target question (Brown et al., 2020; Dong et al., 2024; Wei, Tay, et al., 2022). Few-shot prompting generally has better performance than 0-shot prompting (J. Liu et al., 2022; Min et al., 2022; Zhao, Wallace, Feng, Klein, & Singh, 2021), which only provides instructions. Performance is sensitive to the prompt, including the number of shots (Cao, Law, & Fidler, 2020) and the choice of examples (Zhao et al., 2021).

[1] https://github.com/amrisi/amr-guidelines/blob/master/amr.md

**Chain-of-Thought (CoT) Prompting** Unlike regular few-shot prompting, CoT prompts include not only example question-answer pairs but also intermediate reasoning steps that can derive the final answer (Wei, Wang, et al., 2022). CoT prompting can significantly enhance the capabilities of LLMs in complex reasoning (Lewkowycz et al., 2022; Saparov & He, 2023), and bring more interpretability with the generated reasoning process (Weng et al., 2023). Madaan and Yazdanbakhsh (2022) for instance claim that, through CoT prompting, LLMs can better understand the task by extracting commonsense knowledge from the questions, and generalize to unseen tasks by mimicking the expert's intermediate reasoning steps (Yang, Schuurmans, Abbeel, & Nachum, 2022). However, final answers can be inconsistent with reasoning steps (Lyu et al., 2023).

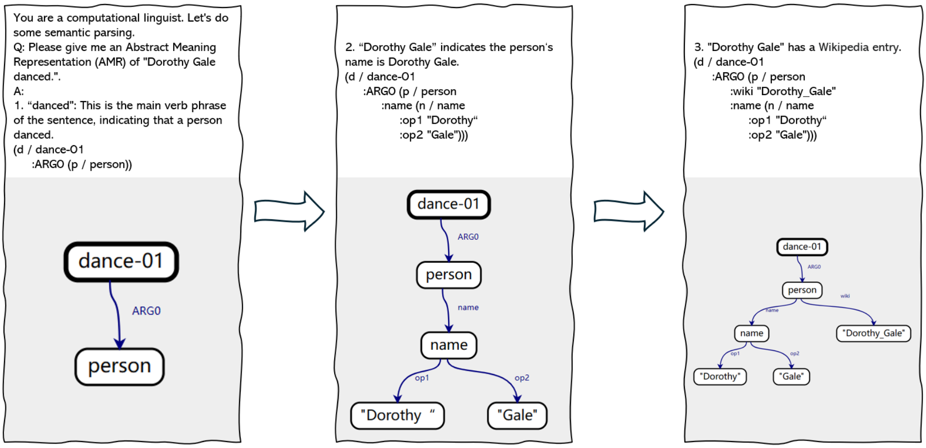



**Figure 1:** A toy example of CoT reasoning (Top-down) for *Dorothy Gale danced.* Each box is a reasoning step, where the top part is the CoT reasoning text and the bottom part is a visualization of the corresponding subgraph

## 3 Experimental Setup and Prompting Strategies

In this paper, we evaluate multiple prompting strategies for generating AMRs using GPT models, including three GPT-3.5 variants—`text-davinci-003`, `gpt-3.5-turbo`, and `gpt-3.5-turbo-instruct` (Brown et al., 2020)—as well as GPT-4o (OpenAI et al., 2024). All experiments are conducted via the official OpenAI API, with temperature set to 0 for reproducibility.[2]

[2]The `gpt-3.5-turbo-instruct` and `GPT-4o` models used correspond to `gpt-3.5-turbo-0613` and `gpt-4o-2024-05-13` at the time of our experiments. Used within terms of use: https://openai.com/policies/eu-terms-of-use/

### 3.1 AMR Dataset

Our experiment was conducted on the English AMR 2.0 (Knight et al., 2017) and AMR 3.0 (Knight et al., 2021) test set with example selections for few-shot prompting on the training set of AMR 2.0 and AMR 3.0, respectively. AMR 2.0 test set has 1,371 AMRs, (test) and 36,521 (train) AMRs, and AMR 3.0 has 1,898 and 55,635. The AMR 2.0 test set is essentially a large subset of that of AMR 3.0. Used within the terms of the license, LDC User Agreement for Non-Members.

### 3.2 Prompting Strategies

All of our prompts request an AMR given a sentence; some include examples. Since role-playing improves model performance (Kong et al., 2023; Reynolds & McDonell, 2021), all prompts begin with *"You are a computational linguist."* We implement and evaluate five prompting strategies: 0-shot, 1-shot, 5-shot, Chain-of-Thought (CoT) with one example, and two-step prompting (details in Appendix A).

1-shot and CoT contain a predefined example sentence: *"The poor kid didn't receive the gift and the postcard that Dorothy Gale sent him on May 25th."*, which was built to demonstrate common AMR properties, such as reentrancy, different non-core roles, etc.

5-shot examples are sampled from the training set using two strategies: random sampling and semantic similarity-based sampling. Using semantically similar examples in prompts can improve LLM performance (Gao, Fisch, & Chen, 2021; J. Liu et al., 2022). We employ Wang et al. (2020)'s model *"sentence-transformers/all-MiniLM-L6-v2"*[3] to compute cosine similarities between the target sentence and training sentences, selecting the top five most similar examples for the prompt.

We introduce two different styles of CoT prompts, top-down (see Figure 1) and bottom-up. The top-down approach begins by identifying the top node, typically the main verb, and subsequently determines its child nodes and their semantic relations in a recursive manner until the complete graph is constructed. The bottom-up approach initially extracts smaller subgraphs, such as the subject, object, location, and time, and then incrementally links these subgraphs through their interrelations until the entire graph is assembled. (See Appendix A.4 for a bottom-up example.)

The two-step prompting strategy combines elements of Chain-of-Thought (CoT) and 5-shot prompting. In Step 1, the model is prompted using 5-shot examples to generate only the core arguments (The nodes linked by labels such as ARG0, ARG1, etc.). In Step 2, a new GPT instance receives the output from Step 1 along with a list of AMR non-core roles (e.g., modifiers) and is prompted to incrementally add non-core roles, guided by a single CoT example.

[3] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

## 4 Results

**Parsability of outputs:** Some LLM output texts are not parsable as AMRs, and some produce multiple AMRs. While the ability of the model to produce a single, valid AMR is the main question of interest, also of interest is whether a pipeline that includes minor post-processing works as an AMR parser. Thus we also apply a post-processing script[4] that fixes mismatched parentheses, splits multiply-labeled nodes into separate nodes, and combines multiple AMRs into one. The proportion of valid AMRs for each method (termed Parsability), before and after post-processing, is in Figure 2.

Without post-processing, in the 0-shot scenario, fully 85% of outputs from GPT -3.5 had syntactic errors rendering them unparsable, and post-processing only brought the parsability rate up to 33%. At the other end of the scale, GPT-4o in the 5-shot scenario was able to generate 86% parsable outputs, and post-processing brought it up to 98%. Thus, left to its own devices, GPT-3.5 at least is entirely unusable, but the more advanced GPT-4o, plus post-processing, is able to generate AMR-formatted text given 5 examples.

**Smatch:** The standard evaluation metric for AMR is **Smatch** (S. Cai & Knight, 2013), which computes the F1 score over the best alignment of triples between the predicted and gold AMR graphs. Each AMR is represented as a set of triples $(s, r, t)$, where $s$ and $t$ are concepts and $r$ is a relation. Smatch is defined as:

$$\text{Smatch} = \frac{2 \times |G_p \cap G_g|}{|G_p| + |G_g|}$$

where $G_p$ and $G_g$ are the sets of triples in the predicted and gold graphs. Here, unparsable graphs were replaced by a dummy graph `(d / dummy)`.

Table 1 shows the Smatch scores of the best version of each method after post-processing, compared with the SOTA AMR parser. Full results, including scores before and after post-processing, are provided in Appendix B.1.

Our best results are for the 5-shot method with GPT-4o (Smatch 60), but nothing approaches the SOTA AMR parser (Vasylenko et al., 2023) with 86.1 on AMR 2.0 and with 84.6 on AMR 3.0. GPT is loosely comparable to an early AMR baseline parser, JAMR, with a 58 Smatch on the original LDC2013E117 AMR dataset (2,100 test sentences; Flanigan, Thomson, Carbonell, Dyer, and Smith (2014)).

The 1-shot and CoT methods provide only one example, and perform poorly, with CoT actually worsening performance (Smatch 36 *vs* 41 with GPT-3.5). The performance of the two-step method was about the same as 5-shot (49 and 50 with GPT-3.5).

Our Smatch results are in keeping with parallel work done by Ettinger et al. (2023), who find that, at best, GPT outputs on the standard AMR 3.0 test set have a Smatch score of around 50.

[4]Script can be found in `https://github.com/liam-0/Fix-ill-formed-AMR.git`

Hand-analysis of CoT sample outputs on AMR 3.0 (Appendix C.1) found a myriad of errors, including mismatches between the reasoning step and partial result. Still, the sampled CoT outputs were, to us, surprisingly good, often making sense and matching the subgraphs generated.

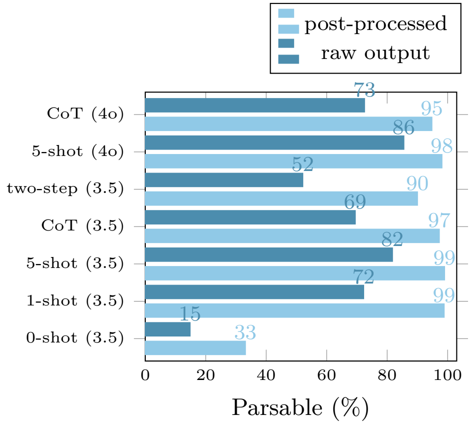



**Figure 2:** Parsability before/after post-processing. The GPT version is indicated in parentheses after each prompting strategy (e.g., CoT (4o))

| **Dataset AMR 3.0** | **Smatch** |
|---|---|
| 5-shot (GPT-4o) | 60 |
| CoT (GPT-4o) | 55 |
| LeakDistill (SOTA) (Vasylenko et al., 2023) | **84.6** |
| **Dataset AMR 2.0** | |
| 0-shot (GPT-3.5) | 14 |
| 1-shot (GPT-3.5) | 41 |
| 5-shot (GPT-3.5) | 50 |
| CoT (GPT-3.5) | 36 |
| two-step (GPT-3.5) | 49 |
| LeakDistill (SOTA) (Vasylenko et al., 2023) | **86.1** |
| **Dataset LDC2013E117** | |
| JAMR (Flanigan et al., 2014) | 58 |

**Table 1:** Smatch for our methods (all post-processed), LeakDistill (SOTA), and JAMR (an early baseline).

## 4.1 Fine-grained results

In addition to Smatch, we evaluated the GPT-4o outputs with the Granular AMR Parsing Evaluation Suite, or GrAPES (Groschwitz et al., 2023), a fine-grained evaluation with 36 categories divided into 9 sets. 23 of the categories are extracted from the AMR 3.0 test set; evaluating with these metrics grants insights into the strengths and weaknesses of GPT as an AMR parser. For comparison, we include a high-performing fine-tuned BART model, AMRBart (Bai et al., 2022) (Smatch 84). Full GrAPES results are in Appendix B.3. We also ran Damonte, Cohen, and Satta (2017)'s fine-grained Smatch on all outputs (see Appendix B.1) and highlight some relevant results here.

Unsurprisingly, AMRBart outperforms GPT in nearly all categories. However, there is substantial and, we argue, principled, cross-categorical variation: overall, GPT is much worse at more complex and AMR-specific tasks. There are also effects of it not having been trained specifically on the AMR training set.

**Seen vs Unseen** Unlike with fine-tuned AMR parsers, GPT shows very little difference in performance on subcategories of things seen and unseen in the AMR 3.0 training set:

| Category | 5-shot 4o | CoT 4o | AMR Bart |
|---|---|---|---|
| Seen vs Unseen | | | |
| Rare node labels | 61 | 57 | 69 |
| Unseen node labels | **61** | 56 | 45 |
| Hard unseen wiki links | **33** | 5 | 9 |
| *Seen* | 71 | 59 | 93 |
| *Unseen* | 57 | 48 | 58 |
| *Seen − Unseen as % of Seen* | 10% | 18% | 38% |
| AMR-specific | | | |
| *PropBank* | 30 | 25 | 63 |
| Multinode word meanings | 14 | 4 | 84 |
| Imperatives | 4 | 0 | 66 |
| Ellipsis | 12 | 15 | 55 |
| *Special Entities* | 64 | 55 | 77 |
| *Average AMR-spec.* | 25 | 20 | 69 |
| *Average all categories* | 48 | 42 | 72 |
| *AMR-spec − all as % of all* | 49% | 53% | 4% |

**Table 2:** Selection of fine-grained categories from GrAPES. *Italicised* categories are averages across multiple categories. Scores are (averages of) recall.

while AMRBart performs on average 38% worse on unseen items, our best GPT model is only 10% worse (Table 2).

**Simple vs complex, AMR-specific subtasks** GPT performs well on simple tasks like node labeling. For instance, fine-grained Smatch includes the F-score over the multiset of node labels, where GPT scores 67, notably higher than its overall Smatch F-score of 60. GPT even outperforms AMRBart on the GrAPES category *Hard unseen wiki links*, which are wiki links for named entities that are not templatic. Evidently, these unpredictable URLs occur in GPT's training data, and it is able to make use of them.

However, the more complex and AMR-specific the subtask, the worse GPT gets. For tasks we classified as AMR-specific (PropBank tasks, multinode word meanings (e.g. *teacher* is annotated `(person :ARG0-of teach-01)`), imperatives, ellipsis, and special entities), GPT performed 49% worse than its average GrAPES score, while AMRBart performed only 4% worse.

**Two-step performance** with GPT-3.5 is comparable to the best 3.5 version (5-shot, Smatch 50 and 49). Here the core roles are predicted in the 5-shot setting, and indeed the fine-grained Smatch score for *SRL* (Semantic Role Labeling, core roles) is equal at 47. Since non-core roles are predicted with CoT and CoT performs worse overall, we might expect overall poor performance for *Negation*, *NER (Named Entity Recognition)*, and *Wiki links*. In fact, performance here is inconsistent, being worse for negation (13 vs 10) and Wiki links (66 vs 59) but identical for NER (69). Also of note is that sampling indicates that step 1 outputs often contain more than just core roles. More insights can be gained here by trying GPT-4o and performing an error analysis on sampled outputs, as we did for CoT.

## 5 Discussion

**GPT performs poorly** LLM performance is correlated with the amount of task-relevant data during pre-training (Kandpal, Deng, Roberts, Wallace, & Raffel, 2023). Even if the whole AMR 3.0 dataset slipped into the GPT training data, it only has 59,255 sentences, yielding near-zero results in the zero-shot setting. The purest version of GPT as an out-of-the-box AMR parser is therefore right out. This is in contrast to, for instance, Python programming, where Poldrack, Lu, and Beguš (2023) found that natural-language prompts for Python code were usable on the first try in 38% of cases. Parsing into Python code is arguably just as difficult a task as AMR parsing, so we might expect similar outcomes were it not for the presumably huge difference in training data quantities.

Fine-grained analysis reveals a large discrepancy between subtasks that are fairly simple and easy to predict, such as basic node labeling, and subtasks that are complex or AMR-specific, such as imperatives. Because many language phenomena have a Zipfian distribution, it is impossible to create a single, short enough example that contains every phenomenon that can – or even is likely to – arise.

A better selection of the examples in the 5-shot may help, since we only need to illustrate phenomena for one sentence at a time. However, the problem of identifying the phenomena to demonstrate, and finding the AMRs that exemplify them, is in itself a kind of parsing. A proof-of-concept experiment could use the gold AMR and measure graph similarity, but this would not be usable as an AMR parsing method.

Despite recent advancements enabling LLMs to process extended contexts (Lin et al., 2024), incorporating AMR annotation guidelines directly into the prompt (as attempted in the second step of our two-step approach) did not yield significant improvements.

Attempts to split the task using a CoT prompting strategy were also unsuccessful. This approach may require multiple CoT examples containing potentially needed non-core roles to form effective k-shot prompts. Moreover, its success is limited not only by the model's context window size but also by the difficulty of obtaining high-quality CoT examples.

**Interpretability of CoT AMR parsing** Although CoT uses a similar setup to one-shot, it is not an extension of the 1-shot method. The output graphs are not necessarily the same, and even when they are, there is no way to know whether the Chain-of-Thought is in any way related to how the model built the graph in the one-shot case.

An advantage of CoT is that it to a certain extent reflects how GPT derives an AMR in the CoT case, since the subgraphs in the chain of reasoning are usually in fact subgraphs of the final output. We can often easily find errors through the CoT reasoning process, which can make it easier to correct errors by hand. The difference between the subgraph and gold AMR could then be used as the loss signal for prompt tuning. This method opens the possibility to subsequently use LLM as a generative model for data augmentation, especially for complex sentences.

## 6 Conclusion

We compared the capabilities of GPT models on AMR parsing under various prompting strategies. We found that GPT-3.5 and GPT-4o make poor AMR parsers, with a maximum Smatch of 60.

Two CoT prompting methods for AMR parsing (bottom-up and top-down) were introduced, as well as a two-step approach, with core and non-core roles added separately. A two-step method was also explored, splitting the task into generating core- and non-core roles. All worsen performance over 5-shot methods, but arguably add some interpretability.

A detailed analysis revealed GPT's limited ability to produce AMRs in alignment with structural AMR guidelines. These guidelines specify precise ways to annotate certain things, such as named entities, that cannot be predicted *a priori*. GPT struggles to express many linguistic phenomena within the AMR paradigm.

## References

Bai, X., Chen, Y., & Zhang, Y. (2022). Graph pre-training for AMR parsing and generation. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 6001–6015). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.acl-long.415` doi: 10.18653/v1/2022.acl-long.415

Blloshmi, R., Tripodi, R., & Navigli, R. (2020). XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 2487–2500). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.195` doi: 10.18653/v1/2020.emnlp-main.195

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`

Cai, D., & Lam, W. (2020). AMR parsing via graph-sequence iterative inference. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1290–1301). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.119` doi: 10.18653/v1/2020.acl-main.119

Cai, S., & Knight, K. (2013). Smatch: an evaluation metric for semantic feature structures. In H. Schuetze, P. Fung, & M. Poesio (Eds.), *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 748–752). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P13-2131`

Cao, T., Law, M. T., & Fidler, S. (2020). A theoretical analysis of the number of shots in few-shot learning. In *International conference on learning representations.* Retrieved from `https://openreview.net/forum?id=HkgB2TNYPS`

Choi, J. D., Bonial, C., & Palmer, M. (2010). Propbank instance annotation guidelines using a dedicated editor, jubilee. In N. Calzolari et al. (Eds.), *Proceedings of the seventh international conference on language resources and evaluation (LREC'10).* Valletta, Malta: European Language Resources Association (ELRA). Retrieved from `http://www.lrec-conf.org/proceedings/lrec2010/pdf/903_Paper.pdf`

Damonte, M., Cohen, S. B., & Satta, G. (2017). An incremental parser for Abstract Meaning Representation. In M. Lapata, P. Blunsom, & A. Koller (Eds.), *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 1, long papers* (pp. 536–546). Valencia, Spain: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/E17-1051`

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., ... Sui, Z. (2024). A survey on in-context learning. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 1107–1128). Miami, Florida, USA: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2024.emnlp-main.64/` doi: 10.18653/v1/2024.emnlp-main.64

Ettinger, A., Hwang, J., Pyatkin, V., Bhagavatula, C., & Choi, Y. (2023). "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the association for computational linguistics: Emnlp 2023* (pp. 8250–8263). Singapore: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.findings-emnlp.553` doi: 10.18653/v1/2023.findings-emnlp.553

Fernandez Astudillo, R., Ballesteros, M., Naseem, T., Blodgett, A., & Florian, R. (2020). Transition-based parsing with stack-transformers. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the association for computational linguistics: Emnlp 2020* (pp. 1001–1007). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.findings-emnlp.89` doi: 10.18653/v1/2020.findings-emnlp.89

Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., & Smith, N. A. (2014). A discriminative graph-based parser for the Abstract Meaning Representation. In K. Toutanova & H. Wu (Eds.), *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1426–1436). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P14-1134` doi: 10.3115/v1/P14-1134

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 3816–3830). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.acl-long.295` doi: 10.18653/v1/2021.acl-long.295

Goodman, M. W. (2020). Penman: An open-source library and tool for AMR graphs. In A. Celikyilmaz & T.-H. Wen (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations* (pp. 312–319). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-demos.35` doi: 10.18653/v1/2020.acl-demos.35

Groschwitz, J., Cohen, S., Donatelli, L., & Fowlie, M. (2023). AMR parsing is far from solved: GrAPES, the granular AMR parsing evaluation suite. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 10728–10752). Singapore: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.emnlp-main.662` doi: 10.18653/v1/2023.emnlp-main.662

Kandpal, N., Deng, H., Roberts, A., Wallace, E., & Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th international conference on machine learning.* JMLR.org. Retrieved from `https://dl.acm.org/doi/10.5555/3618408.3619049`

Knight, K., Badarau, B., Baranescu, L., Bonial, C., Bardocz, M., Griffitt, K., ... Schneider, N. (2017). *Abstract Meaning Representation (AMR) Annotation Release 2.0.* Abacus Data Network. Retrieved from `https://doi.org/10.35111/s444-np87`

Knight, K., Badarau, B., Baranescu, L., Bonial, C., Bardocz, M., Griffitt, K., ... others (2021). *Abstract Meaning Representation (AMR) annotation release 3.0.* https://catalog.ldc.upenn.edu/LDC2020T02. Abacus Data Network.

Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., & Zhou, X. (2023). *Better zero-shot reasoning with role-play prompting.*

Lee, Y.-S., Astudillo, R., Thanh Lam, H., Naseem, T., Florian, R., & Roukos, S. (2022). Maximum Bayes Smatch ensemble distillation for AMR parsing. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 5379–5392). Seattle, United States: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.naacl-main.393` doi: 10.18653/v1/2022.naacl-main.393

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7871–7880). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.703/ doi: 10.18653/v1/2020.acl-main.703

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., … Misra, V. (2022). Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 3843–3857). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf

Lin, B., Zhang, C., Peng, T., Zhao, H., Xiao, W., Sun, M., … Lin, W. (2024). *Infinite-llm: Efficient llm service for long context with distattention and distributed kvcache.* Retrieved from https://arxiv.org/abs/2401.02669

Lindemann, M., Groschwitz, J., & Koller, A. (2020). Fast semantic parsing with well-typedness guarantees. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 3929–3951). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.323 doi: 10.18653/v1/2020.emnlp-main.323

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2022). What makes good in-context examples for GPT-3? In E. Agirre, M. Apidianaki, & I. Vulić (Eds.), *Proceedings of deep learning inside out (deelio 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures* (pp. 100–114). Dublin, Ireland and Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.deelio-1.10 doi: 10.18653/v1/2022.deelio-1.10

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, *55*(9). Retrieved from https://doi.org/10.1145/3560815 doi: 10.1145/3560815

Lyu, Q., Havaldar, S., Stein, A., Zhang, L., Rao, D., Wong, E., … Callison-Burch, C. (2023). Faithful Chain-of-Thought reasoning. In J. C. Park et al. (Eds.), *Proceedings of the 13th international joint conference on natural language processing and the 3rd conference of the asia-pacific chapter of the association for computational linguistics (volume 1: Long papers)* (pp. 305–329). Nusa Dua, Bali: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.ijcnlp-main.20

Madaan, A., & Yazdanbakhsh, A. (2022). *Text and patterns: For effective chain of thought, it takes two to tango.* Retrieved from https://arxiv.org/abs/2209.07686

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp.

11048–11064). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.emnlp-main.759 doi: 10.18653/v1/2022.emnlp-main.759

Naseem, T., Shah, A., Wan, H., Florian, R., Roukos, S., & Ballesteros, M. (2019). Rewarding Smatch: Transition-based AMR parsing with reinforcement learning. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4586–4592). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1451 doi: 10.18653/v1/P19-1451

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., … Zoph, B. (2024). *Gpt-4 technical report.* Retrieved from https://arxiv.org/abs/2303.08774

Peng, X., Gildea, D., & Satta, G. (2018). AMR parsing with cache transition systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/11922 doi: 10.1609/aaai.v32i1.11922

Poldrack, R. A., Lu, T., & Beguš, G. (2023). *Ai-assisted coding: Experiments with GPT-4.* Retrieved from https://arxiv.org/abs/2304.13187

Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 chi conference on human factors in computing systems.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3411763.3451760 doi: 10.1145/3411763.3451760

Saparov, A., & He, H. (2023). Language models are greedy reasoners: A systematic formal analysis of Chain-of-Thought. In *The eleventh international conference on learning representations.* Retrieved from https://openreview.net/forum?id=qFVVBzXxR2V

van Noord, R., & Bos, J. (2017). The meaning factory at SemEval-2017 task 9: Producing AMRs with neural semantic parsing. In S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. Cer, & D. Jurgens (Eds.), *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 929–933). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/S17-2160 doi: 10.18653/v1/S17-2160

Vasylenko, P., Huguet Cabot, P. L., Martínez Lorenzo, A. C., & Navigli, R. (2023). Incorporating graph information in transformer-based AMR parsing. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the association for computational linguistics: Acl 2023* (pp. 1995–2011). Toronto, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.findings-acl.125 doi: 10.18653/v1/2023.findings-acl.125

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 5776–5788). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/

paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., … Fedus, W. (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*. Retrieved from https://openreview.net/forum?id=yzkSU5zdwD (Survey Certification)

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., … Zhou, D. (2022). Chain-of-Thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 24824–24837). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

Weng, Y., Zhu, M., Xia, F., Li, B., He, S., Liu, S., … Zhao, J. (2023). Large language models are better reasoners with self-verification. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the association for computational linguistics: Emnlp 2023* (pp. 2550–2575). Singapore: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.findings-emnlp.167 doi: 10.18653/v1/2023.findings-emnlp.167

Yang, M. S., Schuurmans, D., Abbeel, P., & Nachum, O. (2022). Chain of Thought imitation with procedure cloning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 36366–36381). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/ebdb990471f653dffb425eff03c7c980-Paper-Conference.pdf

Zhang, S., Ma, X., Duh, K., & Van Durme, B. (2019). Broad-coverage semantic parsing as transduction. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3786–3798). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1392 doi: 10.18653/v1/D19-1392

Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 12697–12706). PMLR. Retrieved from https://proceedings.mlr.press/v139/zhao21c.html

Zhou, J., Naseem, T., Fernandez Astudillo, R., & Florian, R. (2021). AMR parsing with action-pointer transformer. In K. Toutanova et al. (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 5585–5598). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.443 doi: 10.18653/v1/2021.naacl-main.443

## A Appendix: Prompt Design Detail

### A.1 0-shot Prompt

```
You are a computational linguist. Let’s do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of ’[Target Sentence]’.
A:
```

### A.2 Few-shot Prompt

#### A.2.1 1-shot Prompt

```
You are a computational linguist. Let’s do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of "The poor kid
didn’t receive the gift and the postcard that Dorothy Gale sent him on May 25th.".
A:
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                              :op1 "Dorothy"
                              :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)
Q: Please give me an Abstract Meaning Representation (AMR) of ’[Target Sentence]’.
A:
```

#### A.2.2 Example 5-shot Prompt

```
You are a computational linguist. Let’s do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of ’However, when it
comes to the soul and the physical being, in the Church’s eyes, the soul trumps
the former.’.
A:
(c2 / contrast-01
      :ARG2 (t / trump-01
            :ARG0 (s2 / soul)
            :ARG1 b
            :ARG1-of (v / view-02
                  :ARG0 (o / organization
                        :wiki "Catholic_Church"
```

```
                         :name (n / name
                             :op1 "Church")))
            :condition (c3 / come-12
                  :ARG1 (a / and
                        :op1 (s / soul)
                        :op2 (b / be-02
                              :mod
                              (p / physical))))))

Q: Please give me an Abstract Meaning Representation (AMR) of 'Thanks for the
article and the background - I read both.'.
A:
(m / multi-sentence
      :snt1 (t / thank-01
            :ARG2 (a / and
                  :op1 (a2 / article)
                  :op2 (b / background)))
      :snt2 (r / read-01
            :ARG0 (i / i)
            :ARG1 (b2 / both)))

Q: Please give me an Abstract Meaning Representation (AMR) of 'International;
weapons; proliferation; dissent; politics.'.
A:
(a / and
      :op1 (i / international)
      :op2 (w / weapon)
      :op3 (p / proliferation)
      :op4 (d / dissent-01)
      :op5 (p2 / politics))

Q: Please give me an Abstract Meaning Representation (AMR) of 'Speakers addressed
opium production and trafficking in Afghanistan.'.
A:
(a / address-01
      :ARG0 (p / person
            :ARG0-of (s / speak-01))
      :ARG1 (a2 / and
            :op1 (p2 / produce-01
                  :ARG1 (o / opium))
            :op2 (t / traffic-01)
            :location (c / country
                    :wiki "Afghanistan"
                    :name (n / name :op1
                            "Afghanistan"))))

Q: Please give me an Abstract Meaning Representation (AMR) of 'I'm sorry for the
swearing, i'm not one to usually swear, but the CIA et al are a joke.'.
A:
(a2 / and
      :op1 (s / sorry-01
            :ARG1 (i / i)
```

```
            :ARG2 (s2 / swear-02
                  :ARG0 i))
      :op2 (s3 / swear-02 :polarity -
            :ARG0 i
            :mod (u / usual)
            :concession-of (j / joke
                  :domain (a / and
                        :op1 (g /
                        government-organization
                        :wiki "Central
                        _Intelligence
                        _Agency"
                            :name (n / name
                            :op1 "CIA"))
                            :op2 (o / other)))))

Q: Please give me an Abstract Meaning Representation (AMR) of 'well, where did
it go in the first place?'.
A:
(g / go-01
      :ARG1 (i / it)
      :ARG4 (a / amr-unknown)
      :prep-in (p / place
            :ord (o / ordinal-entity :value 1)))

Q: Please give me an Abstract Meaning Representation (AMR) of '[Target Sentence]'.
A:
```

### A.3 Top-down CoT Prompt

```
You are a computational linguist. Let's do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of "The poor kid
didn't receive the gift and the postcard that Dorothy Gale sent him on May 25th.".
A:
1. "didn't receive": This is the main verb phrase of the sentence, indicating
that "the poor kid" did not receive something. "the gift and the postcard"
refers to the direct object of the verb "receive", specifying what the poor kid
didn't receive.
(r / receive-01
      :ARG0 (k / kid)
      :ARG1 (a / and)
      :polarity -)
2. "the poor kid" indicates that "kid" is poor.
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and)
      :polarity -)
3. "the gift and the postcard" are the two objects that the poor kid didn't

receive.
(r / receive-01
```

```
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard))
      :polarity -)

4. "that Dorothy Gale sent him on May 25th": This relative clause modifies "the
gift and the postcard" which are sent by a person ("Dorothy Gale") to the kid.
The kid is the recipient of "the gift and the postcard". Hence, there is a
reentrancy at node k. "sent him on May 25th": The action "sent" happened on May
25th, which is a date-entity. The ":month" is 5 and the ":date" is 25.
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person)
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)
5.  "Dorothy Gale sent" indicates the person who sent "the gift and the postcard"
is named "Dorothy Gale".
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                              :op1 "Dorothy"
                              :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)

The AMR is:
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
```

```
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                              :op1 "Dorothy"
                              :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)

Q: Please give me an Abstract Meaning Representation (AMR) of '[Target Sentence]'.
A:
```

### A.4 Bottom-up CoT Prompt

```
You are a computational linguist. Let's do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of "The poor kid
didn't receive the gift and the postcard that Dorothy Gale sent him on May 25th.".
A:
1.  This sentence has the following components: "The poor kid": indicates that
"kid" is poor.
(k / kid
      :mod (p / poor))
"the gift and the postcard": "the gift" and "the postcard" are in an "and"
relationship.
(a / and
      :op1 (g/ gift)
      :op2 (p2 / postcard))
"Dorothy Gale": "Dorothy Gale" is the "name" of a person.
(p3 / person
      :wiki "Dorothy_Gale"
      :name (n / name
            :op1 "Dorothy"
            :op2 "Gale"))
"him": refers to the "The poor kid".
k
"on May 25th": is a date-entity. The ":month" is 5 and the ":date" is 25.
(date-entity
      :month 5
      :day 25)
2. "didn't receive": This is the main verb phrase of the sentence, indicating
that "the poor kid" did not receive something. "the gift and the postcard" is
the direct object of the verb "receive" specifying what the poor kid didn't
receive.
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
```

```
            :op2 (p2 / postcard))
      :polarity -)
3. "that Dorothy Gale sent him on May 25th": The action "sent" is performed by a
person named "Dorothy Gale" to the kid, and it happened on May 25th. The kid is
the recipient of "the gift and the postcard". Hence, there is a reentrancy at
node k.
(s / send-01
      :ARG0 (p3 / person
            :wiki "Dorothy_Gale"
            :name (n / name
                :op1 "Dorothy"
                :op2 "Gale"))
      :ARG1 k
      :time (d / date-entity
            :month 5
            :date 25))
4. "the gift and the postcard that Dorothy Gale sent him on May 25th": "the gift
and the postcard" is the object of "sent". This is a relative clause, so we
make "sent" an "ARG1-of" belonging to "and".
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                            :op1 "Dorothy"
                            :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)
The AMR is:
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                              :op1 "Dorothy"
                              :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
```

```
                              :date 25)))
      :polarity -)
Q: Please give me an Abstract Meaning Representation (AMR) of '[Target Sentence]'.
A:
```

### A.5 Two-step Version: Prompt Design Detail

Since there are hundreds of non-core roles of AMR and prompts have token limitations, it is impossible to provide sufficient numbers of examples in one single sentence or in one CoT prompt.

To address this problem, we break the whole AMR generation process into two parts, firstly generate core roles (e.g. the ARGs) and then add non-core roles in a separate step.

Step 1 is to generate core roles with 5-shot prompting, where the prompts only provide AMR with core roles. There are two different AMR styles of Step 1, one is simply pruning all non-core roles in the examples, which might break an AMR graph into several sub-graphs (notated as "multi-graph"). Another is only pruning the non-core roles leaf nodes, which might be some remnants of non-core roles in the AMR graph, but the AMR graph will remain as a single graph (notated as "one-graph"). A "one-graph" style example is shown in A.5.1.

Step 2 is a CoT prompt with an example and a guideline that includes several non-core roles, an example is shown in A.5.2. The results are shown in Table 7 and Table 9.

#### A.5.1 Step 1: Example 5-shot Prompt for generating core roles

```
You are a computational linguist. Let's do some semantic parsing.
Q: Please give me an Abstract Meaning Representation (AMR) of 'However, when it
comes to the soul and the physical being, in the Church's eyes, the soul trumps
the former.'.
A:
(c2 / contrast-01
      :ARG2 (t / trump-01
            :ARG0 (s2 / soul)
            :ARG1 b
            :ARG1-of (v / view-02
                  :ARG0 (o / organization))
            :condition (c3 / come-12
                  :ARG1 (a / and))))

Q: Please give me an Abstract Meaning Representation (AMR) of 'Thanks for the
article and the background - I read both.'.
A:
(m / multi-sentence
      :snt1 (t / thank-01
            :ARG2 (a / and))
      :snt2 (r / read-01
```

```
            :ARG0 (i / i)
            :ARG1 (b2 / both)))

Q: Please give me an Abstract Meaning Representation (AMR) of 'International;
weapons; proliferation; dissent; politics.'.
A:
(a / and)

Q: Please give me an Abstract Meaning Representation (AMR) of 'Speakers addressed
opium production and trafficking in Afghanistan.'.
A:
(a / address-01
      :ARG0 (p / person
            :ARG0-of (s / speak-01))
      :ARG1 (a2 / and
            :op1 (p2 / produce-01
                  :ARG1 (o / opium))))

Q: Please give me an Abstract Meaning Representation (AMR) of 'I'm sorry for the
swearing, i'm not one to usually swear, but the CIA et al are a joke.'.
A:
(a2 / and
      :op1 (s / sorry-01
            :ARG1 (i / i)
            :ARG2 (s2 / swear-02
                  :ARG0 i))
      :op2 (s3 / swear-02 :polarity -
            :ARG0 i))

Q: Please give me an Abstract Meaning Representation (AMR) of 'well, where did
it go in the first place?'.
A:
(g / go-01
      :ARG1 (i / it)
      :ARG4 (a / amr-unknown))

Q: Please give me an Abstract Meaning Representation (AMR) of '[Target Sentence]'.
A:
```

### A.5.2 Step 2: CoT with guildline

```
You are a computational linguist. Let's do some semantic parsing.
non-core roles list:
:accompanier, :age, :beneficiary, :concession, :condition, :consist-of, :degree,
:destination, :direction, :domain, :duration, :example, :extent, :frequency,
:instrument, :li, :location, :manner, :medium, :mod, :mode, :name, :ord, :part,
:path, :polarity, :polite, :poss, :purpose, :quant,
:range, :scale, :source, :subevent, :time,
:topic,
:unit, :value, :wiki
date-entity:
```

```
:calendar, :century, :day, :dayperiod, :decade, :era, :month, :quarter, :season,
:timezone, :weekday, :year, :year2
conjunctions:
:op1, :op2, :op3, :op4, ...
other components:
:country

Q: Here is the sentence and its half-completed AMR graph.
Sentence:
'The poor kid didn't receive the gift and the postcard that Dorothy Gale sent him
on May 25th.'

Half-completed AMR graph:
(r / receive-01
      :ARG0 (k / kid)
      :ARG1 (a / and
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person)
                  :ARG1 k)))

Please give me an Abstract Meaning Representation (AMR) of the sentence according
to its half-completed AMR graph, and aforementioned non-core roles list.
A:
"(r / receive-01)" requires a ":polarity -" because the receiving "didn't" happen.
(r / receive-01
      :polarity -)

For ":ARG0 (k / kid)", because the "kid" is "poor", it is a modifier. That is the
":mod" in the non-core roles list, so the ":mod" is "(p / poor)".
:ARG0 (k / kid
      :mod (p / poor))

"the gift and the postcard" is a conjunction, so ":ARG1 (a / and)" has "op1" and
"op2" from the list; the ":op1" is "(g/ gift)" and the ":op2" is "(p2 / postcard)"
:ARG1 (a / and
      :op1 (g/ gift)
      :op2 (p2 / postcard))

For ":ARG0-of (s / send-01)", "sent him on May 25th", indicates that the sending
happened on May 25th, which is a time. We use ":time" in the non-core roles list,
and it is a "date-entity"; the ":month" is 5 and the ":day" is 25.
:ARG1-of (s / send-01
      :ARG0 (p3 / person)
      :ARG1 k
      :time (d / date-entity
            :month 5
            :date 25))

For ":ARG0 (p3 / person)",  "Dorothy Gale", is a name, which is ":name" in the
non-core roles list, and also ":wiki" in the non-core roles list, so ":wiki" is
"Dorothy_Gale". For the ":name", the two parts of this name are introduced with
"op1" and "ops2". ":op1" is "Dorothy" and the ":op2" is "Gale".
```

```
:ARG0 (p3 / person
      :wiki "Dorothy Gale"
      :name (n / name
            :op1 "Dorothy"
            :op2 "Gale"))

The AMR is:
(r / receive-01
      :ARG0 (k / kid
            :mod (p / poor))
      :ARG1 (a / and
            :op1 (g/ gift)
            :op2 (p2 / postcard)
            :ARG1-of (s / send-01
                  :ARG0 (p3 / person
                        :wiki "Dorothy_Gale"
                        :name (n / name
                              :op1 "Dorothy"
                              :op2 "Gale"))
                  :ARG1 k
                  :time (d / date-entity
                        :month 5
                        :date 25)))
      :polarity -)

Q: Here is the sentence and its half-completed AMR graph.
Sentence:
'[Target_Sentence]'

Half-completed AMR graph:
[Target_Graph]

Please give me an Abstract Meaning Representation (AMR) of the sentence according
to its half-completed AMR graph, and aforementioned non-core roles list.
A:
```

## B Appendix: Full Results

### B.1 Fine-grained Smatch Results

| Smatch sub-metrics | Definition |
|---|---|
| Unlabeled (Unlab.) | Smatch score after pruning the edge labels. |
| NoWSD | Smatch score which ignores Propbank senses. |
| Concepts (Con.) | F-score on the concept identification task. |
| Named Entity Recognition (NER.) | F-score on the named entity recognition. |
| Negations (Neg.) | F-score on the negation detection. |
| Wikification (Wiki.) | F-score on the wikification. |
| Semantic Role Labeling (SRL.) | Smatch score computed on :ARG-i roles only. |
| Reentrancy (Reen.) | Smatch score on reentrant edges only. |

**Table 3:** Fine-grained Smatch definition (Damonte et al., 2017).[5]

| | | Model | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0-shot | turbo-instruct | 3 | 4 | 3 | 4 | 3 | 0 | 4 | 2 | 4 |
| | | davinci | 6 | 7 | 6 | 7 | 4 | 1 | 4 | 2 | 8 |
| | 1-shot | turbo-instruct | 28 | 34 | 28 | 33 | 27 | 3 | 19 | 17 | 29 |
| | | davinci | 32 | 39 | 33 | 36 | 34 | 9 | 40 | 21 | 33 |
| | 5-shot random | turbo-instruct | 34 | 41 | 35 | **43** | 35 | 3 | 38 | 19 | **35** |
| | | davinci | **37** | **44** | **38** | **43** | 40 | 8 | **48** | 21 | **35** |
| | 5-shot similarity | turbo-instruct | 33 | 39 | 34 | 39 | 41 | 7 | 39 | 20 | 32 |
| | | davinci | **37** | 43 | **38** | 42 | **51** | 10 | **48** | **23** | **35** |
| CoT approach | CoT top-down | turbo-instruct | 14 | 17 | 14 | 15 | 10 | 5 | 7 | 6 | 14 |
| | | davinci | 27 | 32 | 27 | 30 | 12 | 13 | 14 | 16 | 30 |
| | CoT bottom-up | turbo-instruct | 12 | 15 | 12 | 13 | 15 | 7 | 11 | 4 | 12 |
| | | davinci | 24 | 29 | 25 | 28 | 19 | **14** | 16 | 14 | 24 |

**Table 4:** Fine-grained Smatch result of baseline and CoT approach (AMR 2.0, raw output).

| | | Model | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0-shot | turbo-instruct | 5 | 7 | 6 | 7 | 5 | 0 | 1 | 3 | 7 |
| | | davinci | 14 | 18 | 14 | 16 | 10 | 1 | 8 | 8 | 18 |
| | 1-shot | turbo-instruct | 39 | 48 | 40 | 47 | 39 | 6 | 28 | 27 | 40 |
| | | davinci | 41 | 50 | 42 | 47 | 42 | 11 | 51 | 29 | 42 |
| | 5-shot random | turbo-instruct | 42 | 50 | 43 | 52 | 42 | 4 | 47 | 25 | 43 |
| | | davinci | 44 | 52 | 45 | 51 | 48 | 9 | 57 | 26 | 42 |
| | 5-shot similarity | turbo-instruct | 44 | 52 | 46 | 53 | 55 | 10 | 54 | 30 | 43 |
| | | davinci | **50** | **58** | **51** | **57** | **69** | 13 | **66** | **34** | **47** |
| CoT approach | CoT top-down | turbo-instruct | 34 | 44 | 34 | 37 | 23 | 11 | 17 | 22 | 36 |
| | | davinci | 36 | 45 | 37 | 41 | 17 | 14 | 22 | 25 | 40 |
| | CoT bottom-up | turbo-instruct | 30 | 37 | 30 | 34 | 30 | 13 | 24 | 15 | 31 |
| | | davinci | 33 | 41 | 34 | 39 | 26 | **19** | 24 | 22 | 34 |

**Table 5:** Fine-grained Smatch result of baseline and CoT approach (AMR 2.0, post-processed).

[5] https://github.com/mdtux89/amr-evaluation

| Method | Evaluation object | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|
| 5-shot similarity | well-formed AMR only | 61 | 68 | 63 | 69 | 75 | 33 | 71 | 42 | 57 |
| | raw output | 53 | 59 | 54 | 59 | 66 | 28 | 62 | 35 | 49 |
| | post-processed | 60 | 67 | 61 | 67 | 73 | 32 | 69 | 41 | 56 |
| CoT top-down | well-formed AMR only | 58 | 66 | 59 | 65 | 72 | 37 | 29 | 38 | 55 |
| | raw output | 43 | 48 | 44 | 47 | 55 | 27 | 19 | 26 | 39 |
| | post-processed | 55 | 63 | 56 | 62 | 70 | 34 | 25 | 36 | 52 |

**Table 6:** Fine-grained Smatch result on the GPT-4o model (AMR 3.0).

### B.1.1 Two-step Version Result

| Step 1 style | Model name | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|
| one_graph | turbo-instruct | 12 | 14 | 12 | 14 | 16 | 3 | 13 | 7 | 13 |
| | davinci | **24** | **29** | **25** | **28** | **30** | **8** | **25** | **17** | **26** |
| multi_graphs | turbo-instruct | 5 | 5 | 5 | 5 | 4 | 3 | 1 | 2 | 5 |
| | davinci | 11 | 13 | 11 | 13 | 10 | 4 | 5 | 5 | 12 |

**Table 7:** Fine-grained Smatch result of two-step version (AMR 2.0, raw output).

| Step 1 style | Model name | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|
| one_graph | turbo-instruct | 41 | 53 | 42 | 51 | 61 | 13 | 51 | 27 | **47** |
| | davinci | **49** | **59** | **50** | **58** | **69** | **10** | **59** | **39** | **47** |
| multi_graphs | turbo-instruct | 32 | 44 | 33 | 44 | 35 | 12 | 29 | 21 | 41 |
| | davinci | 39 | 50 | 40 | 49 | 50 | 14 | 35 | 31 | 41 |

**Table 8:** Fine-grained Smatch result of two-step version (AMR 2.0, post-processed).

## B.2 Fine-grained result comparison (AMR 2.0)

| | Method | Model | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0-shot | davinci | 6 | 7 | 6 | 7 | 4 | 1 | 4 | 2 | 8 |
| | 1-shot | davinci | 32 | 39 | 33 | 36 | 34 | 9 | 40 | 21 | 33 |
| | 5-shot (similarity) | davinci | **37** | **43** | **38** | **42** | **51** | 10 | **48** | **23** | **35** |
| CoT | CoT top-down | davinci | 27 | 32 | 27 | 30 | 12 | **13** | 14 | 16 | 30 |
| Two-step approach | one_graph | davinci | 24 | 29 | 25 | 28 | 30 | 8 | 25 | 17 | 26 |

**Table 9:** Fine-grained Smatch result comparison among baseline, CoT, and two-step version (AMR 2.0, raw output).

| | Method | Model | Smatch | Unlab. | NoWSD | Con. | NER. | Neg. | Wiki. | Reen. | SRL. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0-shot | davinci | 14 | 18 | 14 | 16 | 10 | 1 | 8 | 8 | 18 |
| | 1-shot | davinci | 41 | 50 | 42 | 47 | 42 | 11 | 51 | 29 | 42 |
| | 5-shot (similarity) | davinci | **50** | 58 | **51** | 57 | **69** | 13 | **66** | 34 | **47** |
| CoT | CoT top-down | davinci | 36 | 45 | 37 | 41 | 17 | **14** | 22 | 25 | 40 |
| Two-step approach | one_graph | davinci | 49 | **59** | 50 | **58** | **69** | 10 | 59 | **39** | **47** |

**Table 10:** Fine-grained Smatch result comparison among baseline, CoT, and two-step version (AMR 2.0, post-processed).

### B.3 GrAPES results

| Set ID | Dataset | Metric | 5-shot | CoT | AMRBart | # |
|---|---|---|---|---|---|---|
| 1 | **Pragmatic reentrancies** | | | | | |
| | Pragmatic coreference (testset) | Edge recall | 08 [03, 22] | 06 [02, 18] | 39 [25, 55] | 36 |
| | | Prerequisites | 19 [10, 35] | 22 [12, 38] | 61 [45, 75] | 36 |
| 2 | **Unambiguous reentrancies** | | | | | |
| | Syntactic (gap) reentrancies | Edge recall | 15 [07, 28] | 27 [16, 42] | 49 [34, 64] | 41 |
| | | Prerequisites | 54 [39, 68] | 39 [26, 54] | 68 [53, 80] | 41 |
| | Unambiguous coreference | Edge recall | 39 [24, 56] | 23 [11, 40] | 65 [47, 79] | 31 |
| | | Prerequisites | 61 [44, 76] | 52 [35, 68] | 77 [60, 89] | 31 |
| 4 | **Rare and unseen words** | | | | | |
| | Rare node labels | Label recall | 61 [57, 64] | 57 [53, 61] | 69 [66, 73] | 676 |
| | Unseen node labels | Label recall | 61 [52, 69] | 56 [47, 65] | 45 [37, 54] | 117 |
| | Rare predicate senses (excl. -01) | Label recall | 21 [13, 34] | 18 [10, 30] | 45 [32, 58] | 56 |
| | | Prerequisites | 82 [70, 90] | 73 [60, 83] | 91 [81, 96] | 56 |
| | Rare edge labels (ARG2+) | Edge recall | 15 [07, 29] | 12 [05, 26] | 35 [22, 50] | 40 |
| | | Prerequisites | 35 [22, 50] | 35 [22, 50] | 72 [57, 84] | 40 |
| 5 | **Special entities** | | | | | |
| | Seen names | Recall | 69 [67, 71] | 71 [69, 73] | 94 [93, 95] | 1788 |
| | Unseen names | Recall | 70 [67, 73] | 67 [64, 70] | 76 [73, 79] | 910 |
| | Seen dates | Recall | 68 [62, 73] | 66 [59, 71] | 94 [90, 96] | 233 |
| | Unseen dates | Recall | 51 [45, 58] | 56 [49, 63] | 86 [81, 90] | 204 |
| | Other seen entities | Recall | 88 [83, 91] | 79 [73, 84] | 97 [94, 99] | 237 |
| | Other unseen entities | Recall | 88 [81, 93] | 70 [61, 78] | 78 [69, 85] | 109 |
| 6 | **Entity classification and linking** | | | | | |
| | Types of seen named entities | Recall | 59 [57, 62] | 61 [58, 63] | 92 [90, 93] | 1628 |
| | | Prerequisites | 67 [64, 69] | 69 [67, 71] | 94 [93, 95] | 1628 |
| | Types of unseen named entities | Recall | 39 [35, 43] | 36 [32, 40] | 51 [47, 55] | 659 |
| | | Prerequisites | 60 [56, 64] | 57 [53, 61] | 70 [66, 73] | 659 |
| | Seen and/or easy wiki links | Recall | 73 [71, 75] | 19 [17, 21] | 87 [85, 88] | 2064 |
| | Hard unseen wiki links | Recall | 33 [28, 39] | 05 [03, 08] | 09 [06, 13] | 277 |
| 7 | **Lexical disambiguation** | | | | | |
| | Frequent predicate senses (incl. -01) | Label recall | 46 [43, 48] | 39 [36, 41] | 86 [84, 88] | 1654 |
| | | Prerequisites | 78 [76, 80] | 73 [70, 75] | 94 [93, 95] | 1654 |
| | Passives | Edge recall | 47 [37, 58] | 28 [19, 38] | 76 [66, 84] | 83 |
| | | Prerequisites | 57 [46, 67] | 39 [29, 49] | 80 [70, 87] | 83 |
| | Unaccusatives | Edge recall | 21 [12, 34] | 27 [17, 41] | 71 [57, 82] | 48 |
| | | Prerequisites | 52 [38, 66] | 48 [34, 62] | 79 [66, 88] | 48 |
| 9 | **Non-trivial word-to-node relations** | | | | | |
| | Ellipsis | Recall | 12 [05, 27] | 15 [07, 31] | 55 [38, 70] | 33 |
| | | Prerequisites | 58 [41, 73] | 45 [30, 62] | 94 [80, 98] | 33 |
| | Multinode word meanings | Recall | 14 [07, 26] | 04 [01, 13] | 84 [71, 92] | 50 |
| | Imperatives | Recall | 04 [01, 11] | 00 [00, 05] | 66 [55, 75] | 76 |
| | | Prerequisite | 66 [55, 75] | 59 [48, 70] | 89 [81, 95] | 76 |

**Table 11:** Results on all GrAPES categories extracted from the AMR 3.0 test set. Grey numbers in square brackets are 95%-Wilson confidence intervals. Our 5-shot and CoT results are for GPT-4o with similarity sampling and top-down methods respectively. AMRBart (Bai et al., 2022) is a competitive fine-tuned BART model.

One GrAPES metric we categorize as AMR-specific deserves special mention: Special Entities, which are things like dates, URLs, and names. GPT does much better here than on other AMR-specific subtasks at 64 (the average of the remaining drops to 15, yielding an average difference from the average GrAPES score of 69%). We suspect GPT succeeds here because this task is very templatic.

## C Appendix: CoT Samples

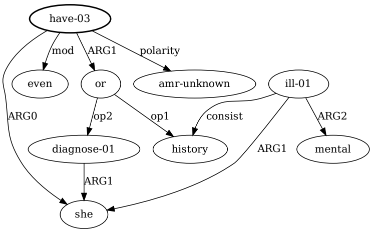
### C.1 Error analysis: Chain-of-Thought

Chain-of-Thought prompting can be taken to lend some interpretability to LLMs. To that end, from the AMR 2.0 test set, we randomly sampled 6 sentences and did an error analysis of the top-down and bottom-up CoT outputs of GPT-3.5. No output was without mistakes; the 96 mistakes noted fall into 10 broad categories (Table 12). To get an idea of what this can look like, consider the bottom-up parse of example (1) in Figure 3. The gold and predicted AMRs are visualized in Figure 4. Steps in bottom-up parses were largely unnumbered, so for ease of exposition, steps are annotated (e.g. [a]).

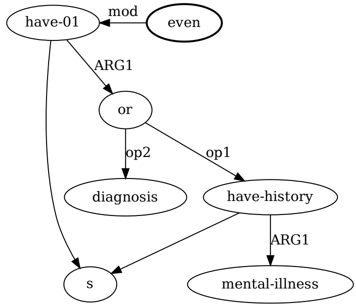
(1) *Does she even have a history of mental illness or a diagnosis?*

```
[a] 1. This sentence has the following components: "Does she": indicates that
"she" is the subject of the sentence.
    (s / she)
[b] "even": indicates that the sentence is asking a question.
    (e / even)
[c] "have": is the main verb phrase of the sentence, indicating that "she" has
something.
    (h / have-01
          :ARG0 s)
[d] "a history of mental illness": indicates that "mental illness" is the object
of the verb "have".
    (h2 / have-history
          :ARG0 s
          :ARG1 (m / mental-illness))
[e] "or a diagnosis": indicates that "diagnosis" is an alternative to "mental
illness".
    (o / or
          :op1 h2
          :op2 (d / diagnosis))
[f] 2. The AMR is:
    (e / even
          :mod (h / have-01
                :ARG0 s
                :ARG1 (o / or
                      :op1 (h2 / have-history
                         :ARG0 s
                         :ARG1
                         (m / mental-illness))
                      :op2 (d / diagnosis))))
```

**Figure 3:** CoT bottom-up output for (1)




**(a)** Gold AMR




**(b)** Predicted AMR

**Figure 4:** AMRs for example (1)

Overall, we can see that the subgraphs generated are not unreasonable, usually make it into the final AMR, and relate at least somewhat to the CoT text. This was true of every item sampled. Step `[a]` is in fact good, with *she* correctly identified and a correct AMR generated for it. All samples started with the correct `ARG0` in AMR.

Step `[b]`, however, is very bad. Not only does *even* have nothing to do with this being a question, but no interrogative marker is added (a Missing Component error). Thus Step `[b]` has two CoT errors: it does not make sense, and the text does not match the subgraph. Step `[c]` is good, only making a PropBank sense error (`have-01` *vs* `-03`). Note as well that the `ARG0` is correctly made node `s`. In Penman notation, only one instance of a node can be labeled as in `(s / she)`; any other mentions of it should just be the node name, here `s`. No outputs we sampled had double-labeled nodes.

In step `[d]` we see two common errors. First, notice the node labeled `mental-illness`, which should be a complex subgraph with `ill-01` and `mental`. We found this kind of "simplification" error 12 times. Node `h2`, labeled `have-history`, is another simplification error, and it is also a CoT error, because the CoT text refers to the verb *have*, but instead of using the existing node `h`, it adds a new node `h2`. Adding a new version of an existing node happened 4 times in the sample, and never was an existing node label updated to reflect the new step's analysis.

In step `[e]`, *diagnosis* should be annotated as a PropBank predicate `diagnose-01`. "Predicate" errors like this occurred 6 times in the sample. Perhaps node `h2` should be thought of as a `history` node; if so, the subgraph dominated by `or` is correct; however, the CoT text here does not match: it should mean `op1` is node `m`, not node `h2`, as it refers to *mental illness*.

In step `[f]`, the subgraphs are put together. Note first that the `ARG0 (s / she)` is missing. Such "Missing earlier component" errors occur 3 times in the sample.

Conversely, the `h2` daughter of `(o / or)` has correctly substituted in the full subgraph dominated by `h2`.

The `even` node has incorrectly been made the root. Its edge to `h` is close: the correct graph has a `mod` edge from `even` to `have-03`.

| Category | BU | TD | Sum |
|---|---|---|---|
| CoT | 10 | 9 | 19 |
| PropBank sense | 4 | 3 | 7 |
| Simplification | 8 | 4 | 12 |
| Predicate | 3 | 3 | 6 |
| Missing component | 8 | 7 | 15 |
| Missing earlier component | 3 | 0 | 3 |
| Named Entity | 4 | 3 | 7 |
| Duplication | 2 | 2 | 4 |
| Whole AMR in step 1 | 0 | 2 | 2 |
| Other | 2 | 18 | 23 |
| Total | 46 | 50 | 96 |
| Steps | 31 | 21 | 52 |

**Table 12:** CoT Error classification over 6 sentences (GPT-3.5; BU = bottom up, TD = top down).

### C.2 Polarity Change Analysis

```
1. This sentence has the following components: "they": refers to a group of
people.
(p / person
      :quant 2)
"not realise": This is the main verb phrase of the sentence, indicating that the
group of people do not realize something.
(r / realize-01
      :ARG0 (p / person
            :quant 2)
      :polarity -)
"that": introduces a subordinate clause.
2. "you don't get piles of change": This is the subordinate clause, indicating
that "you" do not receive a large amount of change.
(r2 / receive-01
```

```
      :ARG0 you
      :ARG1 (c / change
            :quant (p2 / pile)))
"from the bank": specifies where the change is not received from.
(f / from
      :ARG0 c
      :ARG1 (b / bank))
3. "Do they not realise that you don't get piles of change from the bank.": The
main verb phrase is "realise", and the subordinate clause is the object of
"realise". The subordinate clause is also negated, so there is a polarity change
at the top level.
(r / realize-01
      :ARG0 (p / person
            :quant 2)
      :ARG1 (r2 / receive-01
            :ARG0 you
            :ARG1 (c / change
                  :quant (p2 / pile))
            :ARG2 (f / from
                  :ARG0 c
                  :ARG1 (b / bank)))
      :polarity +)
The AMR is:
(r / realize-01
      :ARG0 (p / person
            :quant 2)
      :ARG1 (r2 / receive-01
            :ARG0 you
            :ARG1 (c / change
                  :quant (p2 / pile))
            :ARG2 (f / from
                  :ARG0 c
                  :ARG1 (b / bank)))
      :polarity +)

Golden:
(r / realize-01 :polarity - :mode interrogative
      :ARG0 (t / they)
      :ARG1 (g / get-01 :polarity -
            :ARG0 (y / you)
            :ARG1 (p / pile
                  :mod (c / change))
            :ARG2 (b / bank)))
```

**Figure 5:** CoT sampling: polarity change in ex. (2)

Figure C.2 is the reasoning steps sampled from a top-down CoT generated by turbo-instruct model. The target sentence is in example (2).

(2) *Do they not realise that you don't get piles of change from the bank.*

In the earlier step, GPT added a `(:polarity -)`, because there is a *"not"* in the text. But in the later step, GPT stated that there was another negation, so it needed to switch the polarity. The explanation seems sort of reasonable. This polarity switching is not something AMR does, and was not in the example CoT; GPT 'learned' this by itself. However, here GPT is wrong anyway: the second *not* negates *get*, so there should instead be a second `(:polarity -)` in the AMR.

**Correspondence**

Yanming Li

Inria Saclay, INSA CVL, Paris-Saclay University
PETSCRAFT
Saclay, France
yanming.li@inria.fr

Meaghan Fowlie

Utrecht University
Department of Languages, Literature and Communication
Utrecht, the Netherlands
m.fowlie@uu.nl

Natalia Skachkova, Simon Ostermann, Josef van Genabith, Bernd Kiefer

# Do LLMs fail in bridging generation?


## Abstract

In this work we investigate whether large language models (LLMs) 'understand' bridging relations and can use this knowledge effectively. We present the results obtained from two tasks: generation of texts containing bridging and filling in missing bridging spans. We show that in most of the cases LLMs fail to generate bridging in a reliable way.


## 1 Introduction

Bridging resolution is the task of linking mentions, which are text spans typically representing entities or events, based on some associative relation, such as *part-whole*, *set-member*, *object-attribute*, etc. (Clark, 1975; Poesio, 2004; Poesio, Vieira, & Teufel, 1997). E.g., in the sentence *"The only indication it is* ***a motel*** *is* ***a sign with a faded picture of a locomotive****."* the parts in ***bold*** represent the whole (***a motel***) and its part (***a sign ... locomotive***), and are called an antecedent and an anaphor, respectively.

Bridging resolution is a challenging task - the current state-of-the-art end-to-end bridging resolution model by Kobayashi, Hou, and Ng (2023) reaches maximum 26.2 F1 score on the ISNotes dataset (Markert, Hou, & Strube, 2012). One of the reasons for such a poor performance is a lack of training data - manual annotation of bridging is difficult and costly (Poesio et al., 2018). A potentially promising alternative would be to create more data using an LLM. In this paper we investigate how much LLMs 'know' about bridging and whether they can apply this knowledge to generate new data. Our contributions are two-fold.

- We prompt the *text-davinci-003* model (OpenAI, 2023) to generate 1,000 texts with bridging, and manually investigate in how many of them the relation holds. We show that the model fails to generate texts with bridging in a reliable way.
- We use 13 LLMs to fill in missing bridging antecedents, anaphors, or both of them, and compare the generated spans with the gold ones using a semantic similarity metric. We provide evidence that LLMs have some knowledge of bridging, but often fail to apply it correctly, or 'avoid' using it. We also demonstrate that bridging knowledge contained in LLMs is difficult to extract and quantify.

## 2 Related Work

Investigation of LLMs' capabilities of language 'understanding', as well as estimation of the amount of knowledge they possess, are active research areas. There is evidence that

LLMs have commonsense knowledge (Bubeck et al., 2023; X. L. Li et al., 2022), can infer latent concepts from textual pre-training (Jin & Rinard, 2024) and capture structural semantics (Cheng et al., 2024). On the other hand, Bian et al. (2023); Z. Li et al. (2024); Zhu et al. (2023) and Saba (2024) show that the reasoning and 'understanding' capabilities of LLMs are often exaggerated. While to the best of our knowledge there are no studies on LLMs and bridging relations, there exist some works focusing on the ability of LLMs to capture related phenomena. Dos Santos and Leal (2024), apply different models to assess the strength of semantic similarity between the word pairs, and come to the conclusion that LLMs' predictions correlate with the scores from human annotators. A similar study is conducted by De Deyne, Liu, and Frermann (2024) who use GPT-4 (OpenAI, 2025) to infer semantic relations for human-produced word associations. They find out that the model is good at identification of broad relations, but struggles with more fine-grained ones. Hu, Mahowald, Lupyan, Ivanova, and Levy (2024) investigate the extend to which LLMs can differentiate between grammatical and ungrammatical sentences. They provide evidence that the models' grammaticality judgments align with human intuitions across a range of linguistic phenomena, including anaphora.

## 3 Data

For our study, we use the ARRAU 2 RST corpus (Poesio et al., 2018), as it is one of the largest corpora annotated with bridging relations, and is often used for benchmarking bridging resolution systems. ARRAU 2 RST is a subset of the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) and belongs to the news domain. Dataset statistics and examples of bridging relations can be found in Appendix A.1.

In total, ARRAU 2 RST contains 3,777 bridging pairs. For our experiments, we use the training partition of the dataset and construct the data as follows. We first exclude cases in which the anaphor is part of the antecedent, as we assume that nested spans would be particularly challenging both for the LLM to generate and for us to explain in the prompt. This filtering step yields 2,721 pairs. This subset is used for the first experiment with *text-davinci-003*. Second, for the sake of time and computational efficiency - and to further simplify the task for the models - we limit the number of bridging pairs used in the subsequent experiments. Specifically, we exclude pairs in which the distance between the anaphor and its antecedent exceeds ten whitespace-separated tokens. This results in a set of 554 bridging pairs, which are not necessarily unique. The distribution of bridging relation types in this subset is provided in Appendix A.1. Since each document (or sentence) in ARRAU 2 RST may contain multiple bridging pairs, different pairs may share the same context. To reduce context length, we truncate the text by removing all sentences to the left of the one containing the antecedent and all sentences to the right of the one containing the anaphor. This results in a maximum sequence length of 148 whitespace-separated tokens. Thus, in the second experimental setting, we deliberately focus on bridging spans that occur close to each other in the text. We hypothesize that such spans are significantly easier for LLMs to resolve compared to long-distance and/or nested spans.

Notably, 85 out of the 554 bridging pairs (15.34%) exhibit syntactic head overlap between the antecedent and the anaphor, as in: *"The Labor Department said wage increases in* ***manufacturing industries*** *continue to be smaller than those in* ***other industries*** *."*

We do not know whether ARRAU 2 RST was used in the training of any LLMs. Therefore, we cannot rule out the possibility of data leakage.

## 4 Generating texts with bridging

**Experiment.** We start with an experiment, where we use *text-davinci-003* to generate short texts with bridging. To do the task, the model receives a definition of bridging, an instruction, three demonstrations and a new bridging pair to construct a text with. The demonstrations, as well as the target bridging pair are chosen randomly from the 2,721 ARRAU 2 RST pairs/texts. To identify both antecedent and anaphor in the text, we ask the model to mark them with the "*" symbol on both sides. The prompt is shown in Example 4.1. During text generation, we filter out all texts not following the specified pattern, namely those that have too many or too few "*" symbols. The generation process is executed until we collect 1,000 well-formed texts. Next, we manually check if a bridging relation holds in each text.

**Example 4.1.** *"Bridging is a relation of anaphoric references to non-identical associated antecedents. Bridging covers, for example, part-of, subset, set membership, and possession relations. Make a short text in the style of news with the given words keeping the bridging relation between them.*
*Words:* **** 40 people , or about 15 % **** *and* **** the personnel *.***
*Text: Telxon Corp. said its vice president for manufacturing resigned and its Houston work force has been trimmed by* **** 40 people , or about 15 % **** *. The maker of hand-held computers and computer systems said* **** the personnel **** *changes were needed to improve the efficiency of its manufacturing operation .*
*{two more examples}*
*Words:* **** Federal Reserve banks **** *and* **** branches *.***
*Text:"*

**Results.** Our analysis shows that only 24.4% of all the texts include correct examples of bridging. Another 22.1% represent cases, where the boundaries of the original bridging pairs need to be modified for the bridging relations to hold. The rest (53.5%) do not contain any bridging relations, despite the fact that the given bridging pairs are present in the generated texts. Example 4.2 is a good illustration of the most common problems that occur when using *text-davinci-003* for the task. First, instead of an associative relation between the spans, we have an explicit one (cf. gold text in the same example). Second, the spans' boundaries need to be corrected.

**Example 4.2.** *Gold vs generated texts*
*GOLD: Tenders for the bills , available in minimum $ 10,000 denominations , must be received by 1 p.m. EST Monday at the Treasury or at **Federal Reserve banks** or **branches** .*
*GENERATED: The United States's * **Federal Reserve Banks** * are divided into 12 * **branches** * , each of which holds assets and liabilities of the original Federal Reserve Bank and serves to influence the nation's growth by controlling monetary production and circulation.*

**Discussion.** Although the model's failure in more than half of the cases may be attributed to factors such as suboptimal prompting, inadequate demonstrations, or the inherent difficulty of the task, we hypothesize that the primary reason is that *text-davinci-003* struggles to genuinely 'understand' bridging. As a result, it cannot reliably use bridging in context, even if it may be capable of explaining how two bridging spans are related. As *text-davinci-003* is currently deprecated, we conduct an experiment using *Falcon-40B* to assess whether this may be the case. We extend the prompt in Example 4.1 with an additional instruction requiring the model to provide an explanation of why a bridging relation holds in the generated text. The results indicate that while the model knows the definition of bridging and can explain the relation between two spans, it still frequently fails to generate text that correctly instantiates this relation. The full prompt, along with representative examples of generated texts and explanations, is provided in Appendix A.2. Although *Falcon-40B* cannot be directly compared to *text-davinci-003*, we hypothesize that the latter would likely exhibit similar behavior.

## 5 Fill-in-the-gap task

**Experiments.** To evaluate how well LLMs utilize their knowledge of bridging, we design the following task. For each of the 554 short texts, we successively mask the antecedent, the anaphor, and both spans simultaneously. The LLM is then prompted to process each of the three resulting texts with different types of gaps and to recover the missing spans.

The prompt (see Example 5.1), which is identical across all models, includes four demonstrations. While some LLMs exhibit strong zero- or one-shot capabilities, others may require additional examples to effectively 'understand' the task and produce the desired answer format. Based on our experiments, we found that four demonstrations are optimal for this task. For each masked span or pair, the demonstrations are selected from the remaining 553 gold instances, prioritizing those with the highest semantic similarity to the target spans. To ensure diversity, the spans to be recovered are never identical to those in the demonstrations. Semantic similarity between spans is computed using Sentence-BERT (Reimers & Gurevych, 2019), with similarity scores calculated exclusively on the spans themselves, excluding surrounding context. Notably, the prompt omits both the definition of bridging and any explicit instruction to generate it, as we aim to evaluate how often an LLM can independently infer bridging relations.

**Example 5.1.** *"You are a helpful AI assistant for filling in the gaps in the text. You are given a text containing [MASK] tokens. Replace each [MASK] token with a suitable word.*
***Text with gaps:*** *She also frequently invites directors , producers , actors , [MASK] and [MASK] [MASK] [MASK] [MASK] for coffee and clips .*
***Recovered phrases:***
*writers*
*other show business people*
***Recovered text:*** *She also frequently invites directors , producers , actors , writers and other show business people for coffee and clips.*
*{three more examples}*
***Text with gaps:*** *The show , one of five new [MASK] series , is the second casualty of [MASK] [MASK] [MASK] so far this fall .*
***Recovered phrases:*** *"*

We experiment with publicly available instruct/chat LLMs from different model families and of different sizes, such as *Command* (35B and 104B) (Cohere, 2024), *Falcon* (7B and 40B) (Almazrouei et al., 2023), *Llama3* (8B and 70B) (Grattafiori et al., 2024), *Mistral* (7B and 123B) (Jiang et al., 2023), *Qwen* (7B, 32B and 72B) (Qwen et al., 2025) and *Yi* (9B and 34B) (Young et al., 2025). The full versions are specified in Appendix A.6.

As an exact string match is not suitable for our task, we compare the LLM-generated spans with the original masked spans using a modified version of the BERTScore semantic similarity metric (Zhang, Kishore, Wu, Weinberger, & Artzi, 2020). The rationale for this choice, along with details of the modification, is provided in Appendix A.4. To obtain the lower bounds/baselines, we replace the original bridging pairs with the random, least and most similar pairs (spans) taken from the whole 'pool' of gold bridging pairs in the dataset. Additionally, to assess whether LLMs possess more knowledge about bridging than smaller pre-trained language models, we perform the same recovery task using the encoder-decoder model T5-large (Raffel et al., 2020) and the masked language model DeBERTa-large (He, Liu, Gao, & Chen, 2021).

We formulate the following hypotheses. If an LLM possesses some knowledge about bridging and is able to use it, then 1) the BERT score between the generated spans and the gold ones should be higher than the scores achieved by the baselines and 2) it should be easier for the model to recover one missing span (antecedent or anaphor), than both, i.e. the semantic similarity score should be lower in the latter case.

Importantly, LLMs sometimes produce outputs that do not conform to the format specified in the prompt. For example, a model may generate additional text, return only a single recovered span when two are expected, or omit the recovered spans entirely. When the generated spans cannot be reliably extracted, we insert a dummy span marked as *". . . "* to fill the gap. The number of such invalid outputs produced by each model is reported in Table 5 in Appendix A.5.

**Results.** Figure 1 presents the BERT scores (F1) between predicted and gold spans for three types of gaps. As expected, larger models typically achieve better results. Interestingly, *Qwen* and *Yi* seem to do the task better than other models of similar sizes, with *Qwen-32B* achieving results comparable to those of larger models. All LLMs beat the *DeBERTa-large*, *Random* and *Least-sim* baselines easily, but only really large ones (70B-123B) surpass *T5-large* and can compete with *Most-sim*, especially in the case where both spans are to be restored. The *paired t-test* shows that while 70B-123B LLMs, as well as *Qwen-32B*, reach significantly higher BERT scores than *Most-sim* when recovering missing antecedents and anaphors, the difference to this baseline when restoring both spans is not statistically significant. Thus, our first hypothesis is only partially supported.

Figure 1 shows stronger evidence for the second hypothesis. Namely, for all the LLMs, except *Mistral-7B*, it is more difficult to restore two spans, rather than one, and the difference between the scores is statistically significant. Also, most LLMs tend to struggle more with recovering antecedents, rather than anaphors, which was also confirmed by the paired t-tests. More details can be found in Table 5 in Appendix A.5.

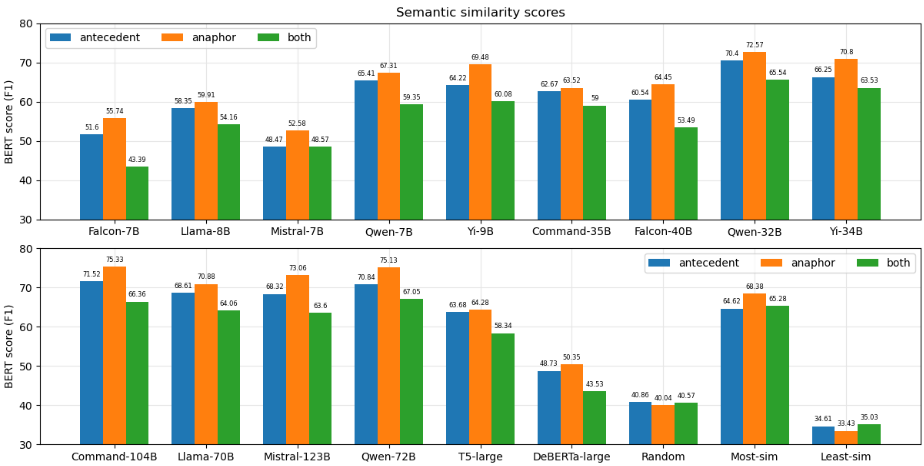



**Figure 1:** BERT scores between gold and predicted spans

To assess model confidence, we compute average perplexity scores for each LLM across 554 texts, evaluating five span types (predicted, gold, most/least similar, random) and three masked slot types (antecedent, anaphor, both). Detailed results appear in Appendix A.7. Perplexity patterns support prior findings: models are less confident when recovering both slots, with random spans yielding the highest perplexities. Predicted and gold spans are rated more probable than others, with predicted spans generally having lower perplexity. Larger models (except *Llama* and *Qwen*) tend to demonstrate lower perplexities than smaller ones in the same family. However, lower perplexity does not always align with higher semantic similarity - for instance, *Yi-9B* has higher perplexity than *Falcon-7B* but achieves better BERT scores.

**Discussion.** While Figure 1 provides evidence that LLMs (at least very large ones) may 'know' what bridging is, and can use this knowledge to some extent, it is also important to note that a high similarity score between generated and gold spans does not necessarily guarantee that the bridging relation is preserved or that the generated text is coherent and grammatically correct. Conversely, a low similarity score does not definitively indicate the absence of a bridging relation in the generated pair.

To investigate whether a high BERT score corresponds to correctly generated bridging pairs, we ask two annotators to manually evaluate 100 randomly selected pairs generated by *Qwen-72B* (one of the best-performing models) and 100 randomly selected pairs produced by *Llama3-8B* (lower-scoring). The annotators assess the number of text sequences in which the bridging relation was preserved. Manual inspection reveals that despite relatively high BERT scores, *Qwen-72B* generates bridging only in 35% of cases on average. Another 9% can be classified as bridging, but have incorrect boundaries (see Example A.10 in Appendix A.3). The performance of *Llama3-8B* is notably lower: bridging relations are found in only 16% of generated pairs, with another 7% potentially classifiable as bridging if the span boundaries were predicted correctly. Inter-annotator agreement, measured by Cohen's Kappa (Cohen, 1960), is 0.41 (moderate) and within the typical range for bridging annotation; for example, Poesio and Vieira (1998) report Cohen's Kappa values between 0.31 and 0.59 for the annotation of definite noun phrases as being in bridging relation or not.

| Prompt | Antecedent | Anaphor | Both |
|---|---|---|---|
| no bridging | 70.84 | 75.13 | 67.05 |
| bridging | 72.42 | 76.32 | 67.53 |

**Table 1:** BERT scores (F1) for 3 types of slots.

To some extent, the low proportion of bridging among the generated spans can be explained by the fact that many masked spans do not necessarily require bridging for the text to be coherent and correct (see Example A.11 in Appendix A.3). Since the prompt does not explicitly instruct the model to generate bridging relations, a model may tend to choose 'easier' candidates to fill in the gaps. To verify this assumption, we repeat the fill-in-the-gap experiment with *Qwen-72B*, augmenting the prompt with a definition of bridging and an explicit instruction to generate bridging. As shown in Table 1, this yields slight improvements; however, the differences are statistically insignificant. Consequently, we conclude that an explicit directive to produce bridging relations does not effectively guide *Qwen-72B* toward the desired behavior. Similar experiments with other models are left for future work.

The models' difficulty in recovering bridging relations may also be influenced by characteristics of the dataset, such as the frequent presence of personal names and numerical expressions as markables, which are challenging for models to reproduce accurately. Additionally, bridging markables and relation types are not consistently defined and vary across datasets (Kobayashi & Ng, 2020). Therefore, we hypothesize that our results may have limited generalizability.

The low number of recovered bridging pairs may also reflect the inherent difficulty of the task. It is well-known that annotating bridging relations is challenging (Poesio et al., 2018; Poesio & Vieira, 1998). However, to our knowledge, no prior studies have investigated human performance on tasks involving filling in missing bridging spans or composing texts based on bridging pairs. For a more rigorous evaluation of LLM capabilities, it would be valuable to compare their performance on these tasks with that of human participants.

Finally, employing the same prompt - albeit concise and simple - for all models may be suboptimal and could contribute to less accurate results. As Mizrahi et al. (2024) highlight, model performance can vary significantly across different instruction paraphrases. Therefore, we plan to conduct a multi-prompt evaluation in future work to ensure robustness.

## 6 Conclusion

In this paper, we investigated to what extent LLMs 'understand' bridging and whether we can use this knowledge for data generation. As our analysis covers only a very small portion of the spans generated by LLMs, it is difficult to draw simple and clear conclusions. Based on the experiments' results, we observe the following trends.

First, bridging remains a highly challenging phenomenon for LLMs, including those with 70B to 123B parameters. Our experiments demonstrate that while such models possess some degree of 'understanding' of bridging, they frequently fail to apply this knowledge effectively. Consequently, their use for reliably generating texts with bridging relations is limited.

Second, measuring bridging is inherently difficult. We observed that many masked gaps can be plausibly filled with non-bridging spans, making it challenging to determine whether an LLM fails due to lack of knowledge or simply opts for simpler candidates. The absence of reliable metrics for identifying bridging further complicates evaluation.

Finally, our preliminary findings require validation on additional bridging datasets, preferably focusing on better-defined subsets of bridging relations. Furthermore, multi-prompt evaluations and comparisons with human performance are necessary to support or refute the trends observed in our initial experiments.

## Acknowledgments

The research in this paper has been funded by the Horizon Europe project Fluently. Grant ID: 101058680.

## References

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., ... Penedo, G. (2023). *The Falcon Series of Open Language Models.* Retrieved from https://arxiv.org/abs/2311.16867

Bian, N., Han, X., Sun, L., Lin, H., Lu, Y., & He, B. (2023). ChatGPT Is a Knowledgeable but Inexperienced Solver: An Investigation of Commonsense Problem in Large Language Models. *ArXiv*, *abs/2303.16421*. Retrieved from https://api.semanticscholar.org/CorpusID:257804619

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... Zhang, Y. (2023). *Sparks of Artificial General Intelligence: Early experiments with GPT-4.* Retrieved from https://arxiv.org/abs/2303.12712

Cheng, N., Yan, Z., Wang, Z., Li, Z., Yu, J., Zheng, Z., ... Han, W. (2024). *Potential and Limitations of LLMs in Capturing Structured Semantics: A Case Study on SRL.* Retrieved from https://arxiv.org/abs/2405.06410

Clark, H. H. (1975). Bridging. In *Proceedings of the 1975 workshop on theoretical issues in natural language processing* (p. 169–174). USA: Association for Computational Linguistics. Retrieved from https://doi.org/10.3115/980190.980237 doi: 10.3115/980190.980237

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46.

Cohere. (2024). *Command R: Retrieval-Augmented Generation at Production Scale.* https://cohere.com/blog/command-r. (Accessed: January, 2025)

De Deyne, S., Liu, C., & Frermann, L. (2024, May). Can GPT-4 Recover Latent Semantic Relational Information from Word Associations? A Detailed Analysis of Agreement with Human-annotated Semantic Ontologies. In M. Zock, E. Chersoni, Y.-Y. Hsu, & S. de Deyne (Eds.), *Proceedings of the workshop on cognitive aspects of the lexicon @ lrec-coling 2024* (pp. 68–78). Torino, Italia: ELRA and ICCL. Retrieved from https://aclanthology.org/2024.cogalex-1.8/

Dos Santos, A. F., & Leal, J. P. (2024). Early Findings in Using LLMs to Assess Semantic Relations Strength (Short Paper). In *Slate.* Retrieved from https://api.semanticscholar.org/CorpusID:274024193

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al. (2024). *The Llama 3 Herd of Models.* Retrieved from https://arxiv.org/abs/2407.21783

He, P., Liu, X., Gao, J., & Chen, W. (2021). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention.* Retrieved from https://arxiv.org/abs/2006.03654

Hu, J., Mahowald, K., Lupyan, G., Ivanova, A., & Levy, R. (2024, August). Language models align with human judgments on key grammatical constructions. *Proceedings of the National Academy of Sciences*, *121*(36). Retrieved from http://dx.doi.org/10.1073/pnas.2400917121 doi: 10.1073/pnas.2400917121

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., ... Sayed, W. E. (2023). *Mistral 7B.* Retrieved from https://arxiv.org/abs/2310.06825

Jin, C., & Rinard, M. (2024). *Latent Causal Probing: A Formal Perspective on Probing with Causal Models of Data.* Retrieved from https://arxiv.org/abs/2407.13765

Kobayashi, H., Hou, Y., & Ng, V. (2023, July). PairSpanBERT: An enhanced language model for bridging resolution. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 6931–6946). Toronto, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.acl-long.383 doi: 10.18653/v1/2023.acl-long.383

Kobayashi, H., & Ng, V. (2020, December). Bridging resolution: A survey of the state of the art. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th international conference on computational linguistics* (pp. 3708–3721). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.331/ doi: 10.18653/v1/2020.coling-main.331

Li, X. L., Kuncoro, A., Hoffmann, J., de Masson d'Autume, C., Blunsom, P., & Nematzadeh, A. (2022, December). A Systematic Investigation of Commonsense Knowledge in Large Language Models. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 11838–11855). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.emnlp-main.812/ doi: 10.18653/v1/2022.emnlp-main.812

Li, Z., Cao, Y., Xu, X., Jiang, J., Liu, X., Teo, Y. S., ... Liu, Y. (2024). LLMs for Relational Reasoning: How Far are We? In *Proceedings of the 1st international workshop on large language models for code* (p. 119–126). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3643795.3648387 doi: 10.1145/3643795.3648387

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330. Retrieved from https://aclanthology.org/J93-2004/

Markert, K., Hou, Y., & Strube, M. (2012, July). Collective Classification for Fine-grained Information Status. In H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, & J. C. Park (Eds.), *Proceedings of the 50th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 795–804). Jeju Island, Korea: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P12-1084/

Mizrahi, M., Kaplan, G., Malkin, D., Dror, R., Shahaf, D., & Stanovsky, G. (2024). State of What Art? A Call for Multi-Prompt LLM Evaluation. *Transactions of the Association for Computational Linguistics*, *12*, 933–949. Retrieved from https://aclanthology.org/2024.tacl-1.52/ doi: 10.1162/tacl_a_00681

OpenAI. (2023). *GPT-3.5 text-davinci-003 model.* https://platform.openai.com. (Accessed: October, 2023)

OpenAI. (2025). *GPT-4.* https://platform.openai.com. (Accessed: February, 2025)

Poesio, M. (2004, July). Discourse annotation and semantic annotation in the GNOME corpus. In *Proceedings of the workshop on discourse annotation* (pp. 72–79). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-0210/

Poesio, M., Grishina, Y., Kolhatkar, V., Moosavi, N., Roesiger, I., Roussel, A., ... Zinsmeister, H. (2018, June). Anaphora resolution with the ARRAU corpus. In M. Poesio, V. Ng, & M. Ogrodniczuk (Eds.), *Proceedings of the first workshop on computational models of reference, anaphora and coreference* (pp. 11–22). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W18-0702 doi: 10.18653/v1/W18-0702

Poesio, M., & Vieira, R. (1998). A corpus-based investigation of definite description use. *Computational Linguistics*, *24*(2), 183–216. Retrieved from https://aclanthology.org/J98-2001/

Poesio, M., Vieira, R., & Teufel, S. (1997). Resolving bridging references in unrestricted text. In *Operational factors in practical, robust anaphora resolution for unrestricted texts.* Retrieved from https://aclanthology.org/W97-1301/

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., ... Qiu, Z. (2025). *Qwen2.5 Technical Report.* Retrieved from https://arxiv.org/abs/2412.15115

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1-67. Retrieved from http://jmlr.org/papers/v21/20-074.html

Reimers, N., & Gurevych, I. (2019, 11). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing.* Association for Computational Linguistics. Retrieved from https://arxiv.org/abs/1908.10084

Saba, W. S. (2024). LLMs' Understanding of Natural Language Revealed. *ArXiv*, *abs/2407.19630*. Retrieved from https://api.semanticscholar.org/CorpusID:271533981

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Rush, A. M. (2020). *HuggingFace's Transformers: State-of-the-art Natural Language Processing.* Retrieved from https://arxiv.org/abs/1910.03771

Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., ... Dai, Z. (2025). *Yi: Open Foundation Models by 01.AI.* Retrieved from https://arxiv.org/abs/2403.04652

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=SkeHuCVFDr

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., ... Zhang, N. (2023). LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities. *World Wide Web (WWW)*, *27*, 58. Retrieved from https://api.semanticscholar.org/CorpusID:258833039

# A Appendix

## A.1 ARRAU 2 RST

All noun phrases in the ARRAU 2 RST dataset are considered markables, which can be referring or non-referring (expletive, quantificational, or predicative), see Table 2 for statistics. Bridging relations are annotated between referring markables and classified into five types, as shown in Table 3. Four of them, namely *possessive*, *subset*, *element* and *other* also have inverse variants. The *undersp-rel* relation is for cases that do not fit into the previous four categories. Examples A.1-A.5 illustrate all five relation types.

| | |
|---|---|
| documents | 413 |
| tokens | 228,901 |
| avg. doc length (tok) | 554.2 |
| markables | 72,013 |
| avg. markables per doc | 174.4 |
| non-referring markables | 9,552 (13.3%) |

**Table 2:** ARRAU 2 RST corpus statistics (from Poesio et al. (2018)).

| | |
|---|---|
| poss / poss-inv | 87 / 25 |
| subset / subset-inv | 1,092 / 368 |
| element / element-inv | 1,126 / 152 |
| other / other-inv | 332 / 7 |
| undersp-rel | 588 |
| total | 3,777 |

**Table 3:** Distribution of bridging in ARRAU 2 RST (from Poesio et al. (2018)).

Table 4 presents the distribution of bridging relation types among the 554 pairs selected for the fill-in-the-gap task. The distribution broadly reflects that of the full dataset, with the notable exception of a higher proportion of relations labeled as *other*. Interestingly, some pairs are annotated with an *unknown* relation, which is not documented in the dataset paper by Poesio et al. (2018).

| | |
|---|---|
| element | 177 (31.95%) |
| other | 124 (22.38%) |
| subset | 101 (18.23%) |
| undersp-rel | 64 (11.55%) |
| subset-inv | 37 (6.68%) |
| poss | 18 (3.25%) |
| element-inv | 16 (2.89%) |
| unknown | 11 (1.99%) |
| poss-inv | 5 (0.90%) |
| other-inv | 1 (0.18%) |

**Table 4:** Distribution of bridging relations in the ARRAU 2 RST subset for the filling-in-the-gap task.

**Example A.1.** *'Possessive' relation*
*Shearson Lehman Hutton Inc. said it applied to Taiwanese securities officials for* ***permission to open brokerage offices in Taipei*** *.* ***Shearson 's application*** *is the first since the Taiwan Securities and Exchange Commission announced June 21 that it would allow foreign brokerage firms to do business in that country .*

**Example A.2.** *'Subset' relation*
***Oil stocks*** *escaped the brunt of Friday 's selling and* ***several*** *were able to post gains , including Chevron , which rose 5 to 66 3 in Big Board composite trading of 2.4 million shares .*

**Example A.3.** *‘Element’ relation*
*Elsewhere in* ***the oil sector*** *,* ***Exxon*** *rallied 7 to 45 3 ;* ***Amoco*** *rose 1 to 47 ;* ***Texaco*** *was unchanged at 51 3 , and* ***Atlantic Richfield*** *fell 1 5 to 99 1 .*

**Example A.4.** *‘Other’ relation*
***The precious metals sector*** *outgained* ***other Dow Jones industry groups*** *by a wide margin for the second consecutive session .*

**Example A.5.** *‘Underspecified’ relation*
*Taiwan officials are expected to review* ***the Shearson application*** *later this year .* ***The new rules*** *will allow investors to buy foreign stocks directly .*

The following characteristics of the ARRAU 2 RST bridging markables are important for our task, as they pose considerable challenges for bridging resolution. First, many of the markables are personal names, e.g., *’Turner Broadcasting System Inc.’*, *’Viacom Pictures’* or *’NBC’*. Some represent an amount of money, like *’53.1 million’* or *’the $200 million portion of the offering’* or other numerical expressions, e.g., *’one brown two-year-old filly’*. Next, the average lengths of an antecedent and an anaphor in the subset used for filling in the gaps task are 4.05 and 3.90 tokens, respectively. However, about 11.01% of the antecedents and 8.84% of the anaphors are longer than 10 tokens. Typically, they contain long relative clauses, e.g., *’Union Carbide, whose third-quarter earnings dropped about 35% from a year earlier and fell short of analysts’ expectations’*.

## A.2 Generating texts with bridging using reasoning

To support the hypothesis that LLMs may ‘know’ the definition of bridging, be able to ‘explain’ the bridging relation between the two spans and still fail to generate a short text where this relation would hold, we present three examples from our experiments with *Falcon-40B*. The prompt we use is based on the prompt for *text-davinci-003* and is given in Example A.6.

**Example A.6.** *"Bridging is a relation of anaphoric references to non-identical associated antecedents. Bridging covers, for example, part-of, subset, set membership, and possession relations. Make a short text in the style of news with the given words keeping the bridging relation between them. Explain why there is a bridging relation between the given words in the text.*
*Words:* **** 40 people , or about 15 % **** *and* **** the personnel ****.
*Text: Telxon Corp. said its vice president for manufacturing resigned and its Houston work force has been trimmed by* **** 40 people , or about 15 % **** *. The maker of hand-held computers and computer systems said* **** the personnel **** *changes were needed to improve the efficiency of its manufacturing operation .*
*Reasoning: There is a bridging relation between * 40 people, or about 15 % * and * the personnel * because * the personnel * introduces a broader group from which the specific subset * 40 people * is later identified.*
*{two more examples}*

*Words: *** Federal Reserve banks *** and *** branches ***.*
*Text:"*

As Examples A.7-A.9 illustrate, *Falcon-40B* typically generates a plausible explanation for why the given spans are in a bridging relation. In Example A.7 the generated text aligns with the explanation, although the span boundaries are slightly incorrect. In Example A.8 the span ** wall ** is missing and bridging relation is not preserved. A similar issue occurs in Example A.9 with the span ** thigh **.

**Example A.7.**
*Words: *** leg *** and *** tripod ****
*Text: The chair was found to have a loose * leg *, causing the customer to trip when they stood up. As a result, the company issued a recall for all * tripods * on the chair.*
*Reasoning: * Leg * and * tripod * are in a bridging relation because * leg * is a specific example of a type of support, and * tripod * is a particular type of leg structure that has three legs.*

**Example A.8.**
*Words: *** wall *** and *** building ****
*Text: Due to the extreme weather conditions, several walls of the old building have collapsed, causing significant damage. Engineers are currently working on plans for repairing and rebuilding the * building *.*
*Reasoning: * Wall * and * building * are in a bridging relation because * wall * is a specific element or part of a building that serves as a support for the structure, while * building * refers to the entire structure.*

**Example A.9.**
*Words: *** thighbone *** and *** thigh ****
*Text: Doctors recommend regular exercise to improve mobility and reduce pain in the * thighbone * and the surrounding areas.*
*Reasoning: * Thighbone * and * thigh * are in a bridging relation, because * thighbone * is a part of the body and * thigh * is a part of the leg, and both parts are identified by their locations.*

### A.3 Fill-in-the-gap task: Typical mistakes

Examples A.10 and A.11 illustrate two typical types of mistakes made by LLMs when performing the fill-in-the-gap task.

**Example A.10.** *Wrong span boundaries*
*PREDICTED: Once inside , she spends nearly four hours measuring and diagramming each room in the 80-year-old house , gathering enough information to estimate what it would cost to rebuild *** it from scratch *** . She snaps photos of *** the original woodwork *** and the plaster that has fallen away from the walls .*
*GOLD: Once inside , she spends nearly four hours measuring and diagramming each room in the 80-year-old house , gathering enough information to estimate what it would*

*cost to rebuild* * ***it*** * *from scratch . She snaps photos of* * ***the original woodwork*** * *and the plaster that has fallen away from the walls .*

**Example A.11.** *Unrelated spans*
*PREDICTED: Early this century , diamond mining in the magnificent dunes where the Namib Desert meets the Atlantic Ocean was a day at* * ***the beach*** * *. Men would crawl in* * ***on hands*** * *looking for shiny stones .*
*GOLD: Early this century , diamond mining in the magnificent dunes where the Namib Desert meets the Atlantic Ocean was a day at* * ***the beach*** * *. Men would crawl in* * ***the sand*** * *looking for shiny stones .*

### A.4 BERT score

The original BERT score compares whole sequences and is not designed to compare their parts. It is possible to extract gold spans and compare them with the predicted ones, but in this case the context, i.e. the surrounding text, will be lost. And if we keep the text, then in most of the cases two sequences will be almost identical and this would lead to BERT score $> 90\%$ no matter what the model predicts. To avoid this problem, we modify BERT score as follows. First, we calculate the contextual embeddings of gold and predicted spans within the original text. Then, we provide span indices to the model and calculate the BERT score only between the embeddings of the spans, masking the embeddings of all the other tokens in the sequence.

### A.5 Invalid generations and T-test statistics

Table 5 reports the proportion of invalid outputs generated by the LLMs. An output is considered invalid if it fails to follow the format specified in the prompt (Section 5), rendering it impossible to extract the recovered phrases.

The table also reveals whether differences in BERT scores achieved by different models for different types of spans are statistically significant. Insignificant differences (i.e. with *p-value* $\geq 0.05$) are given in ***bold***. Given two types of spans, the negative statistic means that the score obtained for the first type is smaller than for the second one, e.g., we see that the BERT scores for the recovered antecedents are smaller than for anaphors across all the models. In most cases, these differences are significant. Next, we compare the scores for antecedents with the scores for both spans. As Table 5 shows, the former are larger than the latter, and the differences are also statistically significant. Given that the scores for antecedents are smaller than for anaphors, we conclude that the differences between the latter and the scores obtained for both spans are significant as well. This supports our hypothesis that for all models it is easier to restore a single bridging span rather than a pair.

| Model | # invalid gen. | antec. vs anaphor | | antec. vs both | |
|---|---|---|---|---|---|
| | | statistic | p-value | statistic | p-value |
| Falcon-7B | 172 (10.35%) | -4.25 | 2.52e-05 | 9.60 | 2.83e-20 |
| Llama-8B | 10 (0.60%) | -1.35 | **0.18** | 4.35 | 1.62e-05 |
| Mistral-7B | 185 (11.13%) | -3.89 | 1.10e-04 | -0.11 | **0.91** |
| Qwen-7B | 39 (2.35%) | -1.58 | **0.12** | 5.92 | 5.66e-09 |
| Yi-9B | 45 (2.71%) | -4.33 | 1.76e-05 | 3.98 | 7.75e-05 |
| Command-35B | 31 (1.87%) | -0.80 | **0.43** | 3.83 | 1.40e-04 |
| Falcon-40B | 239 (14.38%) | -3.57 | 3.90e-04 | 7.39 | 5.34e-13 |
| Qwen-32B | 54 (3.25%) | -1.75 | **0.08** | 4.53 | 7.16e-06 |
| Yi-34B | 148 (8.90%) | -3.78 | 1.70e-04 | 2.56 | 0.011 |
| Command-104B | 42 (2.53%) | -3.06 | 2.00e-03 | 4.58 | 5.82e-06 |
| Llama-70B | 21 (1.26%) | -1.85 | **0.065** | 4.30 | 2.04e-05 |
| Mistral-123B | 98 (5.90%) | -3.68 | 2.60e-04 | 4.09 | 5.02e-05 |
| Qwen-72B | 45 (2.71%) | -3.46 | 5.90e-04 | 3.32 | 9.60e-04 |
| T5-large | 1 (0.06%) | n/a | n/a | n/a | n/a |

**Table 5:** Number of invalid spans (out of 1,662) generated by LLMs and statistical significance of differences in BERT scores (F1) for different types of spans.

### A.6 Models' versions

To save space and memory we use quantized variants of the models [1] from Hugging Face (Wolf et al., 2020).

- `TechxGenus/c4ai-command-r-v01-GPTQ` (35B)
- `alpindale/c4ai-command-r-plus-GPTQ` (104B)
- `tiiuae/falcon-7b-instruct`
- `tiiuae/falcon-40b-instruct`
- `TechxGenus/Meta-Llama-3-8B-Instruct-GPTQ`
- `TechxGenus/Meta-Llama-3-70B-Instruct-GPTQ`
- `TechxGenus/Mistral-7B-Instruct-v0.3-GPTQ`
- `TechxGenus/Mistral-Large-Instruct-2411-GPTQ` (123B)
- `Qwen/Qwen2.5-7B-Instruct-GPTQ-Int8`
- `Qwen/Qwen2.5-32B-Instruct-GPTQ-Int8`

[1] We did not find working quantized *Falcon* models, therefore we use their standard versions.

- `Qwen/Qwen2.5-72B-Instruct-GPTQ-Int8`
- `LnL-AI/Yi-1.5-9B-Chat-4bit-gptq`
- `zgce/Yi-1.5-34B-Chat-GPTQ-Int8`

### A.7 Perplexities

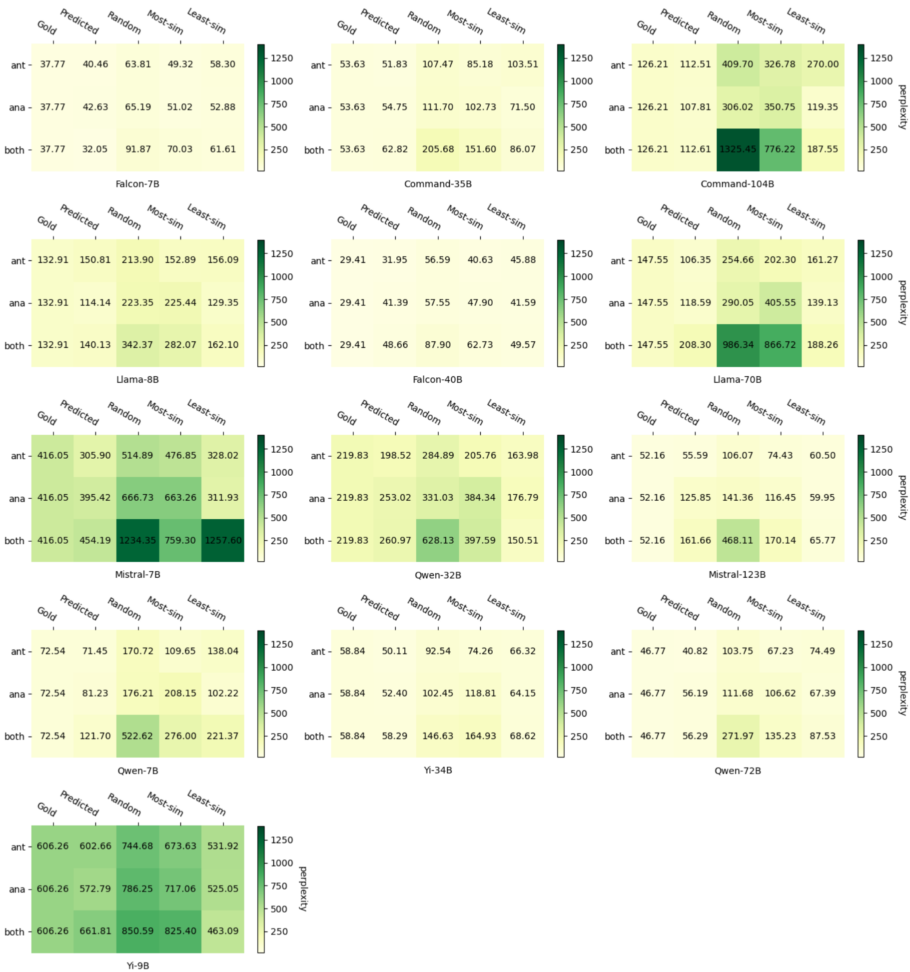



**Figure 2:** LLMs' perplexities for different types of spans

**Correspondence**

Natalia Skachkova

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Saarbrücken, Germany
natalia.skachkova@dfki.de

Sebastian Reimann, Tatjana Scheffler

# The Struggles of Large Language Models with Zero- and Few-Shot (Extended) Metaphor Detection


## Abstract

Extended metaphor is the use of multiple metaphoric words that express the same domain mapping. Although it would provide valuable insight for computational metaphor processing, detecting extended metaphor has been rather neglected. We fill this gap by providing a series of zero- and few-shot experiments on the detection of all linguistic metaphors and specifically on extended metaphors with LLaMa and GPT models. We find that no model was able to achieve satisfactory performance on either task, and that LLaMa in particular showed problematic overgeneralization tendencies. Moreover, our error analysis showed that LLaMa is not sufficiently able to construct the domain mappings relevant for metaphor understanding.


## 1 Introduction

Mappings between an often concrete source domain (e.g. MONEY) and a more abstract target domain (e.g. TIME), so-called conceptual metaphors, structure the way how humans think, according to the Conceptual Metaphor Theory (CMT) of Lakoff and Johnson (1980). These conceptual mappings manifest in language as linguistic metaphors, such as "spending time at home". Extended metaphor, according to Reijnierse, Burgers, Krennmayr, and Steen (2020), represents a special case of linguistic metaphor where multiple metaphor-related words (MRWs) express the same mapping from a source domain to a target domain. Reijnierse et al. (2020) use example (1), taken from a newspaper report on Welsh rugby, to illustrate this. Three MRWs map from the source domain of FIRE to the target domain of RUGBY: a risky arrangement of club fixtures is compared to playing with fire and the negative result is described as being consumed in a conflagration.

(1) They were playing with fire when they decided to arrange a couple of club fixtures and they have been duly consumed in conflagration of their own making.

The automatic detection of single MRWs has received considerable attention within the NLP community. However, extended metaphor remains a phenomenon that has only received little attention in computational work on metaphor. Ge, Mao, and Cambria (2023) explicitly state that there is a lack of work on the detection of extended metaphor, which also extends to the availability of suitable datasets. In such scenarios, zero- (without any labeled examples) and few-shot (with few labeled examples) prompting the current generation of generative, decoder-only large language models (LLMs) like (Chat)GPT and LLaMa has become a useful alternative, as demonstrated in NLP tasks such as sentiment analysis and named entity recognition (Qin et al., 2024).

Ge et al. (2023) moreover note that detecting extended metaphor would require an understanding of domain mappings according to CMT. Consequently, evaluating the performance of LLMs on extended metaphor detection would not just provide insight into the metaphor detection qualities of LLMs but also demonstrate whether the metaphor processing by LLMs follows the assumption made by Lakoff and Johnson (1980) on metaphor and human cognition. We thus make the following contributions:

- We present a series of experiments on metaphor and extended metaphor detection using models from the two most common LLM families and various prompts, where we find that referencing CMT helped overall and that especially LLaMa heavily overgeneralized the positive class. The code for these experiments is publicly available.[1]

- We conduct an extensive error analysis in order to interpret the behavior of LLMs when prompted for extended metaphor detection, which raises serious doubts that the current LLaMa models are able to actually construct domain mappings according to CMT.

## 2 Previous Work

### 2.1 Metaphor and LLMs

Finetuning of encoder-only, pre-trained transformer LMs like BERT has been extensively employed for the task of automatic metaphor detection. Often, multiple encoders were combined to model the linguistic theories of Metaphor Identification Procedure (MIP, Pragglejaz Group, 2007), focusing on a semantic clash between the contextual and a more basic meaning and Selectional Preference Violations (SPV, Wilks, 1975), focusing on the clash between a metaphoric word and its context (Babieno, Takeshita, Radisavljevic, Rzepka, & Araki, 2022; Choi et al., 2021; Li, Wang, Lin, & Guerin, 2023; Zhang & Liu, 2023).

The metaphor identification and interpretation abilities of generative LLMs were so far mostly tested on smaller data. Wachowiak and Gromann (2023) found that GPT-3 was able to predict the source domain of metaphors, mostly from the Master Metaphor List by George Lakoff[2], with an accuracy of 60.22%. Schuster and Markert (2023) included ChatGPT in their zero- and few-shot experiments on metaphor detection in adjective-noun pairs, and found that its zero-shot performance was however outperformed by smaller models that were fine-tuned on labeled data. Goren and Strapparava (2024) tested the ability of GPT-3.5 to identify metaphors in English and Italian proverbs,

[1] https://github.com/SFB-1475/C04-LLMFails-Metaphor
[2] https://www.lang.osaka-u.ac.jp/~sugimoto/MasterMetaphorList/metaphors/index.html

where prompts that asked the model to identify the meaning before identifying the metaphorical parts and the inclusion of larger contexts led to the best results.

The most elaborate approach to metaphor detection via LLMs, called TSI (Theory-Guided Scaffolding Instruction), was put forward by Tian, Xu, and Mao (2024). In order to fill slots in a knowledge graph, TSI prompts GPT-3.5 with a series of questions (either grounded in MIP, SPV or CMT), on the source and target domain of a word and whether these are different. After comparing the structure of the knowledge graphs, TSI provides a label (metaphoric or not). On the TroFi (Birke & Sarkar, 2006) and MOH-X (Mohammad, Shutova, & Turney, 2016) datasets, TSI outperformed several prompting-based methods and BERT models. However, Tian et al. (2024) state that for a large-scale evaluation of their method, they so far lacked the resources.

### 2.2 Computational Approaches to Extended Metaphor

Although the specific automatic identification of extended metaphor in particular has not yet been tackled, some works on metaphor detection have touched upon the concept of extended metaphor. Jang, Maki, Hovy, and Rosé (2017) present a method of automatically finding metaphors that particularly emphasizes extended metaphor. They use an unlabeled corpus and seed words that represent a source domain and its facets (e.g. the domain JOURNEY and *long*) to extract further potential seed words and repeat the procedure several times. Ultimately, features based on these clusters were added to input vectors for an SVM and helped to improve metaphor detection for the JOURNEY domain in posts from a forum of cancer patients.

Reimann and Scheffler (2024) provide a dataset of posts from Christian subreddits annotated via MIPVU and DMIP (Deliberate Metaphor Identification Procedure) (Reijnierse, Burgers, Krennmayr, & Steen, 2018). The latter requires a reason why a metaphorical expression is considered potentially deliberate (i.e. used and intended to be understood "as metaphor"), and a word's status as an extended metaphor was among the possible choices. It comprises annotations for 16,540 tokens, where 3,523 are MRWs, and is further subdivided into a test set of 14,981 tokens and a small training section (used as additional training data in transfer scenarios) of 1,559 tokens. They use the dataset to evaluate the cross-genre transfer capabilities of metaphor detection systems. Additionally, they look at the share of detected potentially deliberate MRWs and find that extended metaphors pose great problems for BERT-based state-of-the-art metaphor detection systems.

## 3 Data and Setup

For evaluation purposes, the test set of Reimann and Scheffler (2024) was a logical choice, given its annotations on extended metaphor. They also provide metaphor annotation on entire Reddit posts and thus larger discourse contexts, which is useful since extended metaphors may stretch over multiple sentences (Reimann & Scheffler, 2024). In total, the dataset contains 281 posts, out of which 72 contain extended metaphor.

To put the results on extended metaphor into the context of general metaphor understanding, we will additionally carry out a token-based identification of all MRWs. This has not yet been tried for the Reddit dataset that we use and, to the best of our knowledge, the detection of metaphoric tokens in text has not been attempted on a larger dataset since both datasets used by Tian et al. (2024) were smaller and focused on metaphoric word pairs.

Including theoretical ideas from CMT into the prompts had a beneficial effect in Tian et al. (2024). For our definition of *extended metaphor*, the notions of source and target domain play a crucial role. Consequently, we design our prompt involving CMT very similarly to the prompt of Tian et al. (2024) without knowledge graphs and scaffolding, which explains the terms metaphor, source domain and target domain according to CMT with the help of an example. In the second part of our prompt, we then define extended metaphor according to Reijnierse et al. (2020). Finally, we also ask the model if an extended metaphor is present and of which MRWs it is made up. We provide all prompts in the appendix.

In our experiments, we use models from the two most frequently used LLM families: LLaMa (Touvron et al., 2023) and GPT (Brown et al., 2020). For LLaMa, we specifically use the instruction-tuned Llama-3.1-8B-Instruct and its larger counterpart Llama-3.1-70B-Instruct, obtained via the HuggingFace transformers library (Wolf et al., 2020). For GPT, we use GPT-4o-mini, which is, at the time of writing, claimed by OpenAI to be more powerful than GPT-3 and GPT-3.5 (used in the approaches mentioned in section 2.1) and advertised as the most cost-efficient version of GPT, which we access via the OpenAI API. The API has a limit of 10,000 requests per day for GPT-4o-mini. Thus, for the word-based, general metaphor detection with GPT, we will thus use only 2,500 tokens from the Reddit dataset. We aim for predictable and reproducible behavior and thus used low values of 0.1 for the top_p and temperature hyperparameters, which control the creativity of the model. For all other hyperparameters, we choose the default values.

## 4 Results

The left side of Table 1 shows the results for the token-based automatic metaphor detection. All models appear to show a large tendency to overgeneralize the metaphor label, with recall much higher than precision. Given the limitations of the OpenAI API and the smaller test set used for the GPT model, a direct comparison of the performance of the two models is not entirely possible. However, it seems that overgeneralization of what may be considered a metaphoric token is much more prominent for LLaMa, compared to GPT. Introducing ideas from CMT had a slight positive effect on precision for all models.

In the extended metaphor detection experiments (right side of Table 1), we observe that the two LLM families exhibit strikingly different behavior. The LLaMa models prompted in a zero-shot fashion exhibit a tendency to overgeneralize by reaching satisfactory levels of recall but largely doing so at the cost of precision. For GPT-4o-

| | Detection of all MRWs | | | | | | Extended Metaphor Detection | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | No CMT | | | CMT | | | Zero | | | Zero+CMT | | | Few+CMT | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| LLaMa 3.1 8b | 23 | 98 | 37 | 25 | 97 | 40 | 38 | 67 | 48 | 36 | 79 | 49 | 46 | 39 | 42 |
| LLaMa 3.1 70b | 26 | 98 | 42 | 33 | 93 | 49 | 46 | 62 | 53 | 41 | 72 | 53 | 46 | 50 | 48 |
| GPT-4o-mini | 32 | 80 | 46 | 42 | 60 | 49 | 53 | 33 | 41 | 56 | 32 | 41 | 59 | 22 | 32 |

**Table 1:** Precision, Recall and F1-scores for the metaphor class in experiments for the detection of all metaphors (left) and extended metaphor (right).

mini, this was the other way around. It was notably more careful in its predictions but failed to recognize a wide range of extended metaphors.

Choosing the 70b version over the 8b version only led to small improvements, suggesting that model size only has a minor impact on the metaphor understanding capabilities of LLaMa 3.1. A notable improvement in precision was achieved by showing the models two examples of extended metaphor. This, however, led vice versa to massive drops in recall, which hints that the models then were not able to properly generalize from the provided examples.

Finally, when evaluating the extraction of the exact MRWs constituting the extended metaphor, the models failed entirely. Only 8% (8b) and 13% (70b) of the MRWs extracted by the LLaMa models were actual MRWs and only 9% of the MRWs extracted by the GPT model were actually labelled as such. Conversely, the LLaMa models also only found 38% (8b) and 31% (70b) of the MRWs in the entire dataset respectively and GPT-4o-mini only detected 18% of the MRWs in the examples containing extended metaphors.

## 5 Error Analysis and Discussion

(2) Before you turn your backyard into a garden or homeless shelter, you need to check city and possibly neighborhood ordinances.

(3) I 've been burned by the hook-up culture many times before . I still have trouble completely renouncing it honestly. What should I do?

(4) Jesus took back the keys of hell at the cross.

In order to better understand the behavior of LLaMa demonstrated in section 4, we had a closer look at the false positives (examples that were falsely classified as extended metaphor) and identified three main categories of errors: (i) overinterpretation of entirely literal text, such as in the non-metaphoric example (2), where LLaMa 8b considered *backyard*, *garden*, *homeless* and *shelter* as MRWs; (ii) MRWs from different

| | Over-interpretation | Different Domains | Wrong Boundaries | Other | Total |
|---|---|---|---|---|---|
| 8b | 44 | 39 | 17 | 2 | 102 |
| 70b | 23 | 33 | 14 | 4 | 74 |

**Table 2:** Distribution of the different error types.

domains, as in (3), where *burned* and *hook-up* were correctly identified as MRWs but express different domain mappings and thus not an extended metaphor; and (iii) wrong boundaries like example (4) that only contains a single MRW (*keys*) but where models recognized an extended metaphor and classified further terms (here: *hell*).

Additionally, for cases that did not completely fit any of the aforementioned categories, we used the category "other issues". Table 2 shows how the different error types are distributed. For the smaller version of LLaMa, the main problem is still its strong tendency to consider a wide range of literal terms to be metaphoric. This changes slightly for the 70B version as the amount of non-metaphoric misclassified examples drops by almost half of the amount of the 8B version. The amount of false positives related to a domain confusion on the other hand remained stable.

This raises doubts whether the models really are able to understand the notion of a domain mapping according to CMT. At first glance, introducing CMT appeared to have improved performance in our experiments. However, the prominence of errors like example (3), that could not even be fixed by choosing a larger model, suggests otherwise. This is partially in line with the findings of Wachowiak and Gromann (2023) for GPT, where the main source of error in predicting the underlying conceptual metaphor was selecting a wrong source domain. In several cases, this happened because GPT-3 was triggered by non-metaphoric words related to the source domain, suggesting again that it lacked the capability to differentiate between domains.

However, it is still hard to discuss the overgeneralizing behavior of LLaMa within the context of previous work since, on the one hand, in previous work on the metaphor understanding capabilities of generative LLMs, mostly models from the GPT family were used. On the other hand, previous studies also employed smaller and more balanced datasets, which may to some extent overshadow such an overuse of the metaphor label as experienced in our case. The Reddit dataset of Reimann and Scheffler (2024) is much larger in size but notably less balanced with only around 20% of the words being metaphors. It can be argued that the large number of non-metaphoric examples is more representative of metaphor use in everyday language (Steen et al., 2010), and thus larger, authentic datasets are more useful for evaluating the metaphor detection and understanding capabilities of LLMs.

Finally, since metaphor annotation is also a challenging task for human annotators, we look at cases where, during the DMIP annotation of Reimann and Scheffler (2024), the two annotators disagreed on extended metaphor. This happened in 40 posts. In 33 of these cases, the annotators decided on the positive label. Seven of them, however, were eventually not considered to express extended metaphor. Out of these seven examples, the two LLaMa models labeled four (8B) and five (70B) as containing an extended metaphor in the best prompting scenario. These overgeneralization cases may thus also be explained by these examples being also ambiguous to human annotators.

## 6 Conclusion and Future Work

We evaluated the capabilities of two state-of-the-art LLM families to find metaphorically used words and extended metaphors. We then carried out a systematic error analysis of the output of the best performing model-prompt-combination. We found that the LLMs failed in two different ways: We observed a general strong bias towards the metaphor and extended metaphor labels, especially with the LLaMa models. Moreover, a closer look at these overgeneralization errors in extended metaphor detection suggests that the models failed to construct the domain mapping required to understand extended metaphor.

Thus, for future work, we suggest to further investigate and find the source of the overgeneralization bias that has plagued all experiments involving LLaMa. Moreover, a more complex prompting approach, similar to for example what Tian et al. (2024) were aiming for, might be worth trying out in order to address the difficulties of the models to understand and connect the source domains of MRWs.

## Funding Statement

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1475 – Project ID 441126958.

## References

Babieno, M., Takeshita, M., Radisavljevic, D., Rzepka, R., & Araki, K. (2022). Miss roberta wilde: Metaphor identification using masked language model with wiktionary lexical definitions. *Applied Sciences*, *12*(4). Retrieved from https://www.mdpi.com/2076-3417/12/4/2081 doi: 10.3390/app12042081

Birke, J., & Sarkar, A. (2006, April). A clustering approach for nearly unsupervised recognition of nonliteral language. In D. McCarthy & S. Wintner (Eds.), *11th conference of the European chapter of the association for computational linguistics* (pp. 329–336). Trento, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/E06-1042

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). *Language models are few-shot learners.* Retrieved from https://arxiv.org/abs/2005.14165

Choi, M., Lee, S., Choi, E., Park, H., Lee, J., Lee, D., & Lee, J. (2021, June). MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In K. Toutanova et al. (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1763–1773). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.naacl-main.141` doi: 10.18653/v1/2021.naacl-main.141

Ge, M., Mao, R., & Cambria, E. (2023). A survey on computational metaphor processing techniques: from identification, interpretation, generation to application. *Artificial Intelligence Review*, 1–67.

Goren, G., & Strapparava, C. (2024, May). Context matters: Enhancing metaphor recognition in proverbs. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)* (pp. 3825–3830). Torino, Italia: ELRA and ICCL. Retrieved from `https://aclanthology.org/2024.lrec-main.338`

Jang, H., Maki, K., Hovy, E., & Rosé, C. (2017, August). Finding structure in figurative language: Metaphor detection with topic-based frames. In K. Jokinen, M. Stede, D. DeVault, & A. Louis (Eds.), *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue* (pp. 320–330). Saarbrücken, Germany: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W17-5538` doi: 10.18653/v1/W17-5538

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* Chicago [u.a.]: Univ. of Chicago Press.

Li, Y., Wang, S., Lin, C., & Guerin, F. (2023, July). Metaphor detection via explicit basic meanings modelling. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 91–100). Toronto, Canada: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.acl-short.9` doi: 10.18653/v1/2023.acl-short.9

Mohammad, S., Shutova, E., & Turney, P. (2016, August). Metaphor as a medium for emotion: An empirical study. In C. Gardent, R. Bernardi, & I. Titov (Eds.), *Proceedings of the fifth joint conference on lexical and computational semantics* (pp. 23–33). Berlin, Germany: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/S16-2003` doi: 10.18653/v1/S16-2003

Pragglejaz Group. (2007, January). MIP: A Method for Identifying Metaphorically Used Words in Discourse. *Metaphor and Symbol*, *22*(1), 1–39. doi: 10.1080/10926480709336752

Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., ... Yu, P. S. (2024). *Large language models meet nlp: A survey.* Retrieved from `https://arxiv.org/abs/2405.12819`

Reijnierse, W. G., Burgers, C., Krennmayr, T., & Steen, G. J. (2018). DMIP: A method for identifying potentially deliberate metaphor in language use. *Corpus Pragmatics*, *2*(2), 129–147.

Reijnierse, W. G., Burgers, C., Krennmayr, T., & Steen, G. J. (2020). The role of co text in the analysis of potentially deliberate metaphor. *Drawing attention to metaphor: Case studies across time periods, cultures and modalities*, 15–38.

Reimann, S., & Scheffler, T. (2024, May). Metaphors in online religious communication: A detailed dataset and cross-genre metaphor detection. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)* (pp. 11236–11246). Torino, Italia: ELRA and ICCL. Retrieved from `https://aclanthology.org/2024.lrec-main.982`

Schuster, J., & Markert, K. (2023, September). Nut-cracking sledgehammers: Prioritizing target language data over bigger language models for cross-lingual metaphor detection. In E. Breitholtz, S. Lappin, S. Loaiciga, N. Ilinykh, & S. Dobnik (Eds.), *Proceedings of the 2023 clasp conference on learning with small data (lsd)* (pp. 98–106). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.clasp-1.12`

Steen, G. J., Dorst, A. G., Herrmann, J. B., Kaal, A. A., Krennmayr, T., & Pasma, T. (2010). *A Method for Linguistic Metaphor Identification: From MIP to MIPVU* (Vol. 14). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Tian, Y., Xu, N., & Mao, W. (2024, June). A theory guided scaffolding instruction framework for LLM-enabled metaphor reasoning. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)* (pp. 7738–7755). Mexico City, Mexico: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2024.naacl-long.428` doi: 10.18653/v1/2024.naacl-long.428

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., … Lample, G. (2023). *Llama: Open and efficient foundation language models.* Retrieved from `https://arxiv.org/abs/2302.13971`

Wachowiak, L., & Gromann, D. (2023, July). Does GPT-3 grasp metaphors? identifying metaphor mappings with generative language models. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1018–1032). Toronto, Canada: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.acl-long.58` doi: 10.18653/v1/2023.acl-long.58

Wilks, Y. (1975). A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, *6*(1), 53–74.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Rush, A. (2020, October). Transformers: State-of-the-art natural language processing. In Q. Liu & D. Schlangen (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-demos.6` doi: 10.18653/v1/2020.emnlp-demos.6

Zhang, S., & Liu, Y. (2023, July). Adversarial multi-task learning for end-to-end metaphor detection. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the association for computational linguistics: Acl 2023* (pp. 1483–1497). Toronto, Canada: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.findings-acl.96` doi: 10.18653/v1/2023.findings-acl.96

## 7 Appendix: LLM Prompts

| Strategy | Prompt |
|---|---|
| CMT | "According to conceptual metaphor theory, metaphor facilitates a mapping of attributes or characteristics from source domain to target domain. For example, the word 'invested' in the sentence 'I have invested a lot of time in her' is a metaphorical expression. The source domain implied by this metaphor is the domain of money and the target domain implied by this metaphor is the domain of time. In the sentence [sentence] decide if the word [word] is a metaphorical expression. If yes, output only label 1, otherwise output only 0." |
| No CMT | "In the sentence {sent} decide if the word {word} is a metaphorical expression. If yes, output only label 1, otherwise output only 0." |

**Table 3:** Prompt templates for the detection of all metaphoric tokens

| Strategy | Prompt |
|---|---|
| Zero | "Extended metaphor represents a particular case of metaphor where several metaphors express the same mapping from a source to a target domain. Based on the above information, decide whether an extended metaphor is expressed in the following text:<br>[post]" |
| Zero+CMT | "According to conceptual metaphor theory, metaphor facilitates a mapping of attributes or characteristics from source domain to target domain. For example, the word 'invested' in the sentence 'I have invested a lot of time in her' is a metaphorical expression. The source domain implied by this metaphor is the domain of money and the target domain implied by this metaphor is the domain of time. Extended metaphor represents a particular case of metaphor where several metaphors express the same mapping from a source to a target domain. Based on the above information, decide whether an extended metaphor is expressed in the following text:<br>[post]" |
| Few-Shot | "According to conceptual metaphor theory, metaphor facilitates a mapping of attributes or characteristics from source domain to target domain. For example, the word 'invested' in the sentence 'I have invested a lot of time in her' is a metaphorical expression. The source domain implied by this metaphor is the domain of money and the target domain implied by this metaphor is the domain of time. Extended metaphor represents a particuar case of metaphor where several metaphors express the same mapping from a source to a domain. This is illustrated in the following two examples:<br>'Another time I heard someone describe Jesus as God ´s character in an MMO. He´s still God , but he´s playing on our server , and Jesus is how we see him in the game.'<br>'Like the closeness between him and God are such that one is the Father and the other is His Son. In this sense it gives a greater meaning to Jesus ( peace be upon him ) and his relationship to God.'<br>In the first example, the words 'character', 'MMO', 'playing', 'server' and 'game' all express a mapping from the source domain of gaming to the domain of religion and transcendence. In the second example, the relationship between God and Jesus is mapped onto the family terms 'Father' and 'Son'. Based on this information, decide whether an extended metaphor is expressed in the following text:<br>[post]" |

**Table 4:** Prompt templates for the detection of extended metaphor

**Correspondence**

Sebastian Reimann

Ruhr-Universität Bochum
Germanistisches Institut
Bochum, Germany
sebastian.reimann@rub.de

Tatjana Scheffler

Ruhr-Universität Bochum
Germanistisches Institut
Bochum, Germany
tatjana.scheffler@rub.de

John David Storment

# Pictorial constituents & the metalinguistic performance of LLMs


### Abstract

In this paper I show that, although ChatGPT (GPT-4o) can provide accurate linguistic acceptability judgments for many types of sentences (Cai, Duan, Haslett, Wang, & Pickering, 2024; Collins, 2024a, 2024b; Ortega-Martín et al., 2023; Wang et al., 2023), it does not give accurate grammaticality judgments for sentences that contain pro-text emojis, which are emojis that appear in a written utterance as morphosyntactic constituents (Cohn, Engelen, & Schilperoord, 2019; Pierini, 2021; Storment, 2024; Tieu, Qiu, Puvipalan, & Pasternak, 2025, a.o.). I demonstrate this with three distinct experiments performed on GPT-4o using both English and Spanish data. This work builds on prior research that shows that the combinatorics of pro-text emojis are sensitive to the morphosyntactic constraints of the language in which the emojis appear, and it connects the poor performance of GPT-4o in this respect to two factors: (i) the fact that, while LLMs are able to make some generalizations of syntactic structural dependencies, their mechanisms for making such generalizations are not derived in the same way that human syntactic structures are (Contreras Kallens, Kristensen-McLachlan, & Christiansen, 2023; Hale & Stanojević, 2024; Kennedy, 2025; Linzen & Baroni, 2021; Manova, 2024a, 2024b; Zhong, Ding, Liu, Du, & Tao, 2023, a.o.), and (ii) the fact that LLMs lack the means of directly processing iconic and pictorial content in the same way that human cognition allows for. I also consider the possibility that the relevant data are poorly attested in the model's training parameters. This paper establishes a precedent for the research of the intersection of generative AI and utterances that contain pictorial elements as morphosyntactic constituents.


### 1 Introduction

ChatGPT is very good at giving human-like acceptability judgments for many English sentences (for some preliminary studies, see Collins (2024a, 2024b)). In some ways, ChatGPT's language use very closely resembles that of humans (Cai et al., 2024; Ortega-Martín et al., 2023; Wang et al., 2023); however, there are many ways in which ChatGPT and other LLMs like it fail to linguistically perform on the same level as humans (Basmov, Goldberg, & Tsarfaty, 2024; Borji, 2023; Jang & Lukasiewicz, 2023; Shen, Chen, Backes, & Zhang, 2023; Zhong et al., 2023; Zuccon & Koopman, 2023, a.o.). These findings suggest that, while ChatGPT and other LLMs are able to generate human-like linguistic utterances, they do not have the same mechanism of generating internal hierarchical syntactic structures that human language is endowed with (Contreras Kallens et al., 2023; Hale & Stanojević, 2024; Linzen & Baroni, 2021; Manova, 2024b; H. Zhou et al., 2023, a.o.).

Emojis are pictorial elements that exist in the digital keyboard layout of most smartphones and computers. They are easily integrated with text and, as such, appear often in written language in computer-mediated communication (CMC). Emojis often appear in written utterances as morphosyntactic constituents Storment (2024), a phenomenon known as pro-text emojis. There is strong evidence to suggest that pro-text emojis are fully integrated into linguistic systems in terms of their morphosyntax combinatorics (Homann, Brady R. T., Sara, & Fernandes, 2022; Stamatov, 2017; Storment, 2024), processing (Paggio & Tse, 2022; Scheffler, Brandt, Fuente, & Nenchev, 2022; Weissman, Engelen, Baas, & Cohn, 2023), and their mechanisms of semantic interpretation (Cohn, Roijackers, Schaap, & Engelen, 2018; Tieu et al., 2025). Emojis are unique as an incredibly widespread, conventionalized set of symbols and pictograms that can be readily incorporated into written natural language. In this paper I refer specifically to pictorial symbols accessible from a digital keyboard layout and associated with a specific Unicode point as "emojis", so this definition excludes traditional "emoticons" composed of preexisting text symbols such as :) and :(. These elements offer a very meaningful glimpse at the way that human language interfaces with visual information.

In this paper I present novel data from GPT-4o (OpenAI et al., 2024) handling sentences containing pro-text emojis in English and in Spanish. I show that, while GPT-4o is able to accurately identify what many – but crucially not all – emojis mean, it is completely unable to predict with any accuracy where pro-text emojis can acceptably appear in written utterances. This is the first study of its kind, as research on the morphosyntactic combinatorics of pro-text emojis is incredibly limited, and research on the metalinguistic awareness of LLMs giving grammaticality judgments is still relatively new at the time of authorship of this paper. Given these considerations, this article outlines a preliminary case study of how one LLM (GPT-4o) handles a relatively small sample of data from two languages (both of which have abundant data online) and how it shows notable differences from the way in which it conveys metalinguistic acceptability judgments on sentences containing pro-text emojis when compared to how it rates sentences which lack emojis. Future work on this topic should address similar questions using a significantly larger sample size of data from significantly more languages with tests run on multiple models, not just GPT-4o.

Human cognition allows for the extraction of meaning from visual stimuli, and this seems to be strongly connected to the human language faculty when we look at the existence of language phenomena such as iconicity, partial iconicity, pictorial symbols in orthography, gesture, and, of course, the existence of signed languages. ChatGPT is crucially missing that aspect of the human capacity for language, and it seems to have no functional alternative, even for visual elements that appear in text. This gap – along with the gaps in syntactic awareness, which has been well-documented – spells doom for LLMs when it comes to the treatment of visual elements as syntactic constituents. This is a gap that must be filled if generative AI models are to truly match human language performance.

## 2 Pro-text emojis

Pro-text emojis (borrowing terminology from the semantics of gestures in spoken language (Schlenker, 2019)) are emojis that appear inside of an utterance as morphosyntactic constituents. See the following examples from Storment (2024), ultimately from Twitter/X.

(1) a. I need to 🛌 before I see the end of this game or I'll be 😢 I missed it

b. 🏠 is where the ❤️ is

c. Some 🌈 people were discriminated against at protest grounds

This phenomenon is not restricted to English. See the following examples from Spanish, also taken from Twitter/X.

(2) a. Mejor me voy a 😴...
better me go.1sg to 😴...
'I'd better go to sleep...'

b. Y que se vayan a tu país los que odian el 🇵🇾
and that REFL go.3pl.sbj to your country those that hate the 🇵🇾
'And that those who hate Paraguay go to your country'

c. Mi solidaridad con la gente 🏳️‍⚧️ y no-binaria siempre!
my solidarity with the people 🏳️‍⚧️ and non-binary always!
'My solidary with trans and non-binary people always!'

Some work on the semantics of pro-text emojis suggest that they replace written words in an utterance (Tieu et al., 2025), but this conclusion is puzzling as pro-text emojis may appear as elements smaller than a completely formed word in a given derivation in both English and in Spanish. Examples again from Twitter/X.

(3) a. He ❤️s to 📚

b. My therapist 👻ed me so I took selfies in the parking lot

c. likeeee the secondhand embarrassment is ☠️ing me

d. A couple of 🚬s smoking 🚬s

(4) a. Las 🍓s me encantaaan
the.fpl 🍓pl me love.3pl
'I looove strawberries'

b. Un ☕cito para reconfortar
a.msg ☕DIM for refresh
'A little coffee to refresh you'

It is not the case, though, that pro-text emojis may freely replace any morpheme. Storment (2024) systematically demonstrates that there are licit and illicit morphosyntactic positions in which pro-text emojis are licensed, and language users have clear grammaticality judgments about what these positions are.

(5) a. *I like 🌸al perfumes (int: floral)

b. *Professor Rambow is a 💻ational linguist (int: computational)

c. *🤨ity killed the cat (int: curiosity)

d. *I need to 💪en my core (int: strengthen)

The examples in (5) are ungrammatical because such forms are unattested online and because English-speaking emoji users consistently judge these forms to be ungrammatical.

There are also restrictions on pro-text emoji placement in other languages, though the restrictions vary from language to language. Take the example of Spanish.

(6) a. Yo te ❤️(*-o) mucho
I you ❤️(*-1SG.PRES) much
'I love you very much'

b. Tú me ❤️(*-s) mucho
you me ❤️(*-2SG.PRES) much
'You love me very much'

While verbal agreement and tense suffixes are possible with pro-text emojis in English (2a-c), they are generally not possible in Spanish. This shows that the combinatorics of pro-text emojis with other morphemes are sensitive to the morphosyntactic structure of the language in which they are embedded. See Storment (2024) for a detailed analysis of this difference between English and Spanish.

These data clearly show two very important facts. First, pro-text emojis do not replace orthographic words. They appear as units smaller than the word-level, yet it is not the case that they can freely stand in for any morpheme. As such, they take part in morphosyntactic operations and must be obedient to the grammatical constraints of the language in which they appear, which is the second important fact.

## 3 GPT-4o tests

I used OpenAI's ChatGPT interface to perform a series of linguistic experiments with GPT-4o on sentences containing pro-text emojis. I performed these experiments on two languages: English and Spanish. I tested GPT-4o's knowledge on emoji recognition in isolation, emoji recognition embedded in sentences, and metalinguistic acceptability judgments on sentences containing pro-text emojis. I detail the findings of these short experiments here.

### 3.1 Emoji recognition

GPT-4o is, for the most part, exceptionally good at recognizing emojis. This is consistent with prior research (Y. Zhou, Lu, Gao, Mei, & Ai, 2024). It can identify what an emoji depicts, how it is used, and even what an emoji's Unicode code sequence is. I asked GPT-4o to identify what a given emoji depicts, what its keywords are, and what its Unicode entry is for a random sample of 40 emojis, five from each category of emoji: Smileys and People, Animals & Nature, Food & Drink, Activity, Travel & Places, Objects, Symbols, and Flags. I ran this prompt (shown in (7), followed by a numbered list of forty emojis) three times.

(7) *Emoji recognition prompt*
Identify the meaning, keywords, and Unicode code associated with the following emojis.

GPT-4o was able to identify all information correctly for all emojis in all three tests. In other words, it performed perfectly.

However, during some preliminary testing in which I gave GPT-4o the prompt in (8), some of the results were quite different.

(8) *Preliminary testing for emoji recognition*
What is the metadata for the emoji _____? I mean the Unicode entry and keywords that are associated with each emoji.

I did this for thirteen random emojis, and it did not perform perfectly. It failed to identify three emojis: 🦷 (tooth), 👂 (ear), and 🦵 (leg). In addition to misidentifying the 🦷 emoji, the Unicode code it gave here is for the safety pin (🧷) emoji. At the time of writing this, there currently is no pigeon emoji. The Unicode code for the 👂 emoji given here was correct, despite misidentifying the meaning. GPT 4-o identified the giraffe emoji as a "monkey face emoji", and, while there is a monkey face emoji that exists, the Unicode code given was the one for the giraffe (🦒) emoji. GPT-4o is mostly able to correctly identify emojis, but it still makes mistakes. The mistakes seem inconsistent and unpredictable, though I was able to force it to misidentify these three emojis several more times outside of the context of this formal experiment. Interestingly, the three emojis that I found GPT-4o to struggle with all depict a part of the human body.

### 3.2 Translation

For the next task, I had GPT-4o "translate" sentences containing pro-text emojis into sentences that only contain standard orthographic words. Using both grammatical and ungrammatical forms, I gave GPT-4o the following prompt along with 40 sentences containing pro-text emojis.

(9) *Translation prompt*
Paraphrase each of the following sentences without using emojis.

In the prompt, I did not indicate which sentences were grammatical and ungrammatical. GPT-4o executed this task perfectly, and accurately conveyed the meaning – or indented meaning – of each sentence. This study demonstrates that GPT-4o has a solid grasp on the semantic content of emojis and their conventions of use, though there is no indication here that it has any notion of syntax or at least morphological combinatorics. I confirm this in the following experiment.

### 3.3 Acceptability judgments

I then had GPT-4o rate grammatical and ungrammatical utterances for their acceptability. I did this for 50 English sentences and 35 Spanish sentences containing pro-text emojis. I ran each prompt three times. I found that GPT-4o is inaccurate and inconsistent when it comes to giving grammaticality judgments of sentences containing pro-text emojis, and these judgments do not match those from human native speakers.

I gave GPT-4o the following prompt(s). I had to specify "informal English/Spanish" because otherwise it judged almost every sentence to be unacceptable.

(10) *Acceptability judgment prompt*
For each sentence, tell me if it is acceptable in informal English/Spanish or not. Do not give any explanations.

One reviewer points out that "acceptable" in the context of this prompt is a very general term. Acceptable could mean anything from logically acceptable to politically acceptable. While this is true, the fact that other studies (e.g. Collins, 2024a, 2024b) as well as preliminary experimentation for this project use the term acceptability to refer to human-like intuitions about the grammaticality of certain linguistic forms, and that this is crucially something that the models seem to grasp in these experiments as they give accurate answers, is at least somewhat indicative that the models understand what acceptability refers to in this context.

Preliminary data from Collins (2024a, 2024b) show that GPT-4o is quite good at giving native-speaker-like grammaticality judgments for sentences that do not contain pro-text emojis, and my data here show that GPT-4o's judgments are quite inaccurate for emoji sentences. I assigned each sentence in each language a match score, which was determined by how many times GPT-4o accurately gauged the acceptability of each sentence across all three trials (i.e., a match score of 66.6667 indicates that it gave an accurate response 2/3 times). Here I show some examples of grammatical and ungrammatical forms in both English and Spanish that were used in this experiment, as well as the GPT-4o match score for each of these sentences in parentheses.

(11) a. I ❤️ you (100)
b. He is the 🏳️‍🌈est person ever (100)
c. I love 🥭es (0)
d. Wow, 👊 must really love yourself (100)

(12) a. *My son is learning ➕ition in school (66.6667)
b. *Mr. Kamano is 🇯🇵ese (66.6667)
c. *Wow, you must really love 👊self (33.3333)

(13) a. Te ❤️ mucho (100)
b. Aquí están los 🐶itos (100)
c. Las 🍎s son rojas (66.6667)

(14) a. *Tú me ❤️s mucho (0)
b. *Las 🍎as son rojas (100)
c. *Yo te ❤️é antes (33.3333)

Acceptability (i.e., grammatical sentences) for these input sentences was determined by attestation online, as well as being judged as grammatical by native speakers of each language in medium-scale judgement tasks with 10-15 participants. In these tasks, native speakers of each language were simply asked to rate sentences as either grammatical or ungrammatical. Interestingly, Spanish speakers felt more strongly about the ungrammaticality of the ungrammatical examples than did English speakers, an observation mirrored in Storment (2024). These data are supported by the lack of attestation for ungrammatical forms. Sentences shown here marked as grammatical were unanimously ranked as such, and the same is true for the ungrammatical utterances. In other words, the sentences in (11) and (13) are marked as acceptable because they are forms that native speakers produce and they are forms that native speakers say are grammatical. Ungrammatical examples such as those in (12) and (14) are marked as such because they are unattested online and because native speakers of English and Spanish report that these sentences are not grammatical. Judgements from human language users are clear on utterances containing pro-text emojis. Judgments from both humans and LLMs on sentences which do not contain pro-text emojis are also clear. This experiment demonstrates that LLM judgments on sentences containing pro-text emojis are what is least clear.

The average match score for all English sentences is 63.3332, and the average match score for all Spanish sentences is 69.5238. GPT-4o performed slightly better with the

Spanish examples than it did with the English examples. Interestingly, this is parallel with the observation in Storment (2024) that native Spanish speakers have more robust judgments on sentences containing pro-text emojis in Spanish than do English speakers in English. More research and experimentation is necessary to confirm if there is actual correlation there, or if it is just coincidence.

Overall, GPT-4o gave inaccurate ratings for these sentences. It especially struggled to label the ungrammatical examples as such, though it also inaccurately labeled many grammatical sentences. In the following section I discuss the theoretical implications of this both for human language and for generative AI systems.

## 4 Theoretical implications

Emoji combinatorics – that is, where an emoji may appear within a given utterance – reveal a hierarchical internal syntax that these emojis must abide by. There is some internal hierarchical structure in each language that licenses these visual elements in some locations, but not others. The distribution of these elements forces us to consider the syntactic structures of natural language.

LLMs, however, do not operate over hierarchically-ordered syntactic constituents in the exact same way that human language does (Manova, 2024a, 2024b), and may not do so at all with pictorial constituents. This raises two important questions. First, what parameters does the LLM use to determine the grammaticality of these utterances containing pro-text emojis? Second, how does it determine the grammaticality of any utterance? The idea is that humans use the same metrics to determine the grammaticality of the emoji utterances than they would any other, but it is not clear that that is the case for LLMs, especially given how relatively rare these kinds of data are. LLMs like GPT-4o determine the grammaticality of utterances by generalizing over the data in their training sets. If a given form frequently appears in the system's training data, it is more likely to accept such forms as grammatical. Conversely, if a given form is poorly attested or completely unattested in the training data, it is less likely to accept those forms.

Frequency and attestation is one of the metrics used in Storment (2024) to determine the acceptability of a given pro-text emoji form. In that paper, grammatical forms are well-attested online (in addition to being judged grammatical by native speakers), and ungrammatical forms are either poorly attested or not attested (also in addition to being judged ungrammatical by native speakers). As shown in the numerous studies cited in the introduction of this paper, the training data works particularly well in many cases to cause the LLM to come to the correct conclusions on grammaticality, though it is not perfect. In fact, the metrics of attestation and frequency are quite accurate in Storment (2024) for determining the grammaticality of emoji sentences, so one might expect a greater degree of accuracy from the LLMs in this way. This forces us to wonder why GPT-4o struggles particularly with these data containing pro-text emojis, assuming it contains such utterances in its training set. If it does not, then this is an obvious way in which the model can improve, though I assume it does have some exposure to this kind of data because it can interpret the sentences successfully and because the model is trained on 175 billion parameters. It could, of course, also be the case that these forms are not in the training data, or at least not enough for the model to meaningfully abstract over the data. This would explain why GPT-4o struggles with some grammatical (i.e., attested) examples as well. This would make sense given that, of all the written utterances online, utterances containing emojis are relatively infrequent. Furthermore, pro-text emojis, despite being used by many language users, are perhaps the most infrequently-occurring use of emojis when compared to things like pro-text and co-text emojis (Storment, 2024; Tieu et al., 2025). It could very well be the case that GPT-4o simply isn't very familiar with these forms, but I consider other options here as to maximally inform a theory of LLM language use and human language cognition.

Generative AI systems do not have the capacity for processing iconic and pictorial elements, at least not directly. Obviously, they do not have the means of directly perceiving and processing visual and auditory stimuli in the way that that humans do. Whatever iconicity is (Davidson, 2023), and however the human brain processes that information, LLMs lack that same resource. One has to wonder if the improvement of an LLM's ability to process visual information would correlate with an improvement in the model to accurately gauge the acceptability of sentences containing pictorial constituents.

Human language readily integrates iconic and pictorial elements, be it depictive sounds (Wiese, 1996), gestures (Goldin-Meadow & Brentari, 2017; Schlenker, 2019), emojis (Grosz, Kaiser, & Pierini, 2021; Storment, 2024), or a more traditional notion of "pictures". Storment (2024) shows that the way in which these elements are incorporated into language is indicative of some underlying internal structure (this idea is expressed elsewhere, such as Wiese (1996)). The ability to incorporate iconic visual elements into language perhaps relies on the ability of these elements to be embedded in a syntax. In other words, the distribution and combinatorics of pictorial morphosyntactic constituents in language reveals certain considerations on the morphosyntactic structures of the languages that these elements appear in. Given that the hierarchical internal structure that LLMs assign to utterances is not done in the same way that humans do it (Contreras Kallens et al., 2023; Hale & Stanojević, 2024; Linzen & Baroni, 2021; Manova, 2024a, 2024b; Y. Zhou et al., 2024, a.o.), and given that the distribution of pictorial elements in linguistic utterances is both dependent on and indicative of hierarchical internal structure, it stands to reason that GPT-4o struggles with giving human-like acceptability judgments for sentences containing pro-text emojis when it normally does not struggle with such a thing is because it is unable to make human-like generalizations about the internal syntactic structure of these pictorial elements specifically. The potential lack of attestation of these forms in GPT-4o's training data – along with the lack of a human-like syntax – could explain why the model is unable to generalize over these data containing pro-text emojis.

## 5 Conclusion

The use of emojis and other pictorial elements is an informatic tool, a tool that forces us to consider the morphosyntactic constraints of a given language. We can use this tool to see where the limitations of LLMs' comprehension lie.

In this paper I show that the LLM GPT-4o cannot make accurate generalizations about the grammaticality of visual elements that are embedded inside utterances in English and Spanish, even though it, for the most part, can very accurately interpret the semantic content of emojis and the utterances in which they appear.

This paper is meant to lay the foundations for future research on this subject. The data and theoretical discussion presented here are still very preliminary, and they introduce many interesting questions concerning iconicity in syntax, human cognition, and the improvement of generative AI.

## References

Basmov, V., Goldberg, Y., & Tsarfaty, R. (2024, April). *Simple Linguistic Inferences of Large Language Models (LLMs): Blind Spots and Blinds.* arXiv. Retrieved 2025-06-03, from `http://arxiv.org/abs/2305.14785` (arXiv:2305.14785 [cs]) doi: 10.48550/arXiv.2305.14785

Borji, A. (2023, April). *A Categorical Archive of ChatGPT Failures.* arXiv. Retrieved 2025-06-03, from `http://arxiv.org/abs/2302.03494` (arXiv:2302.03494 [cs]) doi: 10.48550/arXiv.2302.03494

Cai, Z. G., Duan, X., Haslett, D. A., Wang, S., & Pickering, M. J. (2024, March). *Do large language models resemble humans in language use?* arXiv. Retrieved 2025-06-03, from `http://arxiv.org/abs/2303.08014` (arXiv:2303.08014 [cs]) doi: 10.48550/arXiv.2303.08014

Cohn, N., Engelen, J., & Schilperoord, J. (2019, August). The grammar of emoji? Constraints on communicative pictorial sequencing. *Cognitive Research: Principles and Implications*, *4*(1), 33. Retrieved 2025-06-03, from `https://doi.org/10.1186/s41235-019-0177-0` doi: 10.1186/s41235-019-0177-0

Cohn, N., Roijackers, T., Schaap, R., & Engelen, J. (2018). Are emoji a poor substitute for words? {Sentence} processing with emoji substitutions. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 40). Retrieved from `https://escholarship.org/uc/item/33d802h8`

Collins, C. (2024a, October). *Ordinary Working Grammarian: Acceptability Judgments in ChatGPT.* Retrieved 2025-06-03, from `https://ordinaryworkinggrammarian.blogspot.com/2024/10/acceptability-judgments-in-chatgpt.html`

Collins, C. (2024b, October). *Ordinary Working Grammarian: Acceptability Judgments in ChatGPT (Study 2).* Retrieved 2025-06-03, from `https://ordinaryworkinggrammarian.blogspot.com/2024/10/acceptability-judgments-in-chatgpt_20.html`

Contreras Kallens, P., Kristensen-McLachlan, R. D., & Christiansen, M. H. (2023). Large Language Models Demonstrate the Potential of Statistical Learning in Language. *Cognitive Science*, *47*(3), e13256. Retrieved 2025-06-03, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13256 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13256) doi: 10.1111/cogs.13256

Davidson, K. (2023). Semiotic distinctions in compositional semantics. *Proceedings of the 58th Meeting of the Chicago Linguistic Society.* Retrieved from https://projects.iq.harvard.edu/files/meaningandmodality/files/davidson2023-clsproceedings-semiotic.pdf

Goldin-Meadow, S., & Brentari, D. (2017, January). Gesture and language: Distinct subsystem of an integrated whole. *Behavioral and Brain Sciences*, *40*, e74. Retrieved 2025-06-03, from https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/abs/gesture-and-language-distinct-subsystem-of-an-integrated-whole/66110F96CE4668BD5FD0918BE5E36258 doi: 10.1017/S0140525X1600039X

Grosz, P., Kaiser, E., & Pierini, F. (2021, September). Discourse anaphoricity and first-person indexicality in emoji resolution. *Proceedings of Sinn und Bedeutung*, *25*, 340–357. Retrieved 2025-06-16, from https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/941 doi: 10.18148/sub/2021.v25i0.941

Hale, J. T., & Stanojević, M. (2024, November). Do LLMs learn a true syntactic universal? In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 17106–17119). Miami, Florida, USA: Association for Computational Linguistics. Retrieved 2025-06-03, from https://aclanthology.org/2024.emnlp-main.950/ doi: 10.18653/v1/2024.emnlp-main.950

Homann, L. A., Brady R. T., R., Sara, A., & Fernandes, M. A. (2022, April). Are emojis processed visuo-spatially or verbally? Evidence for dual codes. *Visual Cognition*, *30*(4), 267–279. Retrieved 2025-06-03, from https://doi.org/10.1080/13506285.2022.2050871 doi: 10.1080/13506285.2022.2050871

Jang, M. E., & Lukasiewicz, T. (2023, November). *Consistency Analysis of ChatGPT.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2303.06273 (arXiv:2303.06273 [cs]) doi: 10.48550/arXiv.2303.06273

Kennedy, M. (2025, May). Evidence of Generative Syntax in LLMs.. Retrieved 2025-06-03, from https://openreview.net/forum?id=v2w8LUpt0a

Linzen, T., & Baroni, M. (2021, January). Syntactic Structure from Deep Learning. *Annual Review of Linguistics*, *7*(Volume 7, 2021), 195–212. Retrieved 2025-06-03, from https://www.annualreviews.org/content/journals/10.1146/annurev-linguistics-032020-051035 (Publisher: Annual Reviews) doi: 10.1146/annurev-linguistics-032020-051035

Manova, S. (2024a, June). *ChatGPT and linguistic theory, with a focus on morphology.* LingBuzz. Retrieved 2025-06-03, from https://ling.auf.net/lingbuzz/008600 (LingBuzz Published In: Submitted for inclusion in José-Luis Mendívil-Giró (ed.), Artificial Knowledge of Language. A Linguists' Perspective on its Nature, Origins and Use. Wilmington, DE: Vernon Press.)

Manova, S. (2024b, November). *Machine learning versus human learning: Basic units and form-meaning mapping.* LingBuzz. Retrieved 2025-06-03, from https://lingbuzz.net/lingbuzz/008548 (LingBuzz Published In: Submitted for inclusion in Vsevolod Kapatsinski and Gašper Beguš (eds.), Implications of Neural Networks and Other Learning Models for Linguistic Theory. Special issue of Linguistics Vanguard.)

OpenAI, Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., … Malkov, Y. (2024). *GPT-4o System Card.* arXiv. Retrieved 2025-06-20, from https://arxiv.org/abs/2410.21276 (Version Number: 1) doi: 10.48550/ARXIV.2410.21276

Ortega-Martín, M., García-Sierra, O., Ardoiz, A., Álvarez, J., Armenteros, J. C., & Alonso, A. (2023, February). *Linguistic ambiguity analysis in ChatGPT.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2302.06426 (arXiv:2302.06426 [cs]) doi: 10.48550/arXiv.2302.06426

Paggio, P., & Tse, A. P. P. (2022). Are Emoji Processed Like Words? An Eye-Tracking Study. *Cognitive Science*, *46*(2), e13099. Retrieved 2025-06-03, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13099 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13099) doi: 10.1111/cogs.13099

Pierini, F. (2021, September). Emojis and gestures: a new typology. *Proceedings of Sinn und Bedeutung*, *25*, 720–732. Retrieved 2025-06-16, from https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/963 doi: 10.18148/sub/2021.v25i0.963

Scheffler, T., Brandt, L., Fuente, M. d. l., & Nenchev, I. (2022, February). The processing of emoji-word substitutions: A self-paced-reading study. *Computers in Human Behavior*, *127*, 107076. Retrieved 2025-06-03, from https://www.sciencedirect.com/science/article/pii/S074756322100399X doi: 10.1016/j.chb.2021.107076

Schlenker, P. (2019, May). Gestural semantics. *Natural Language & Linguistic Theory*, *37*(2), 735–784. Retrieved 2025-06-03, from https://doi.org/10.1007/s11049-018-9414-3 doi: 10.1007/s11049-018-9414-3

Shen, X., Chen, Z., Backes, M., & Zhang, Y. (2023, October). *In ChatGPT We Trust? Measuring and Characterizing the Reliability of ChatGPT.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2304.08979 (arXiv:2304.08979 [cs]) doi: 10.48550/arXiv.2304.08979

Stamatov, E. G. (2017). *Do emoji use a grammar? Emergent structure in non-verbal digital communication* (Master's thesis, Tilburg University). Retrieved from https://arno.uvt.nl/show.cgi?fid=142877

Storment, J. D. (2024, May). Going  lexicon? The linguistic status of pro-text emojis. *Glossa: a journal of general linguistics*, *9*(1). Retrieved 2025-06-03, from https://www.glossa-journal.org/article/id/10449/ (Number: 1 Publisher: Open Library of Humanities) doi: 10.16995/glossa.10449

Tieu, L., Qiu, J. L., Puvipalan, V., & Pasternak, R. (2025, April). Experimental evidence for a semantic typology of emoji: Inferences of co-, pro-, and post-text emoji. *Quarterly Journal of Experimental Psychology*, *78*(4), 808–826. Retrieved 2025-06-03, from https://doi.org/10.1177/17470218241255786 (Publisher: SAGE Publications) doi: 10.1177/17470218241255786

Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., … Xie, X. (2023, August). *On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2302.12095 (arXiv:2302.12095 [cs]) doi: 10.48550/arXiv.2302.12095

Weissman, B., Engelen, J., Baas, E., & Cohn, N. (2023). The Lexicon of Emoji? Conventionality Modulates Processing of Emoji. *Cognitive Science*, *47*(4), e13275. Retrieved 2025-06-03, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13275 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13275) doi: 10.1111/cogs.13275

Wiese, R. (1996, January). Phrasal compounds and the theory of word syntax. *Linguistic Inquiry*, *27*, 183–193.

Zhong, Q., Ding, L., Liu, J., Du, B., & Tao, D. (2023, March). *Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2302.10198 (arXiv:2302.10198 [cs]) doi: 10.48550/arXiv.2302.10198

Zhou, H., Hou, Y., Li, Z., Wang, X., Wang, Z., Duan, X., & Zhang, M. (2023, November). *How Well Do Large Language Models Understand Syntax? An Evaluation by Asking Natural Language Questions.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2311.08287 (arXiv:2311.08287 [cs]) doi: 10.48550/arXiv.2311.08287

Zhou, Y., Lu, X., Gao, G., Mei, Q., & Ai, W. (2024, April). *Emoji Promotes Developer Participation and Issue Resolution on GitHub.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2308.16360 (arXiv:2308.16360 [cs]) doi: 10.48550/arXiv.2308.16360

Zuccon, G., & Koopman, B. (2023, February). *Dr ChatGPT, tell me what I want to hear: How prompt knowledge impacts health answer correctness.* arXiv. Retrieved 2025-06-03, from http://arxiv.org/abs/2302.13793 (arXiv:2302.13793 [cs]) doi: 10.48550/arXiv.2302.13793

**Correspondence**

John David Storment

Stony Brook University
Department of Linguistics
Stony Brook, NY, USA
john.david.storment@stonybrook.edu

Simon Münker

# Political Bias in LLMs: Unaligned Moral Values in Agent-centric Simulations


## Abstract

Contemporary research in social sciences increasingly utilizes state-of-the-art generative language models to annotate or generate content. While these models achieve benchmark-leading performance on common language tasks, their application to novel out-of-domain tasks remains insufficiently explored. To address this gap, we investigate how personalized language models align with human responses on the Moral Foundation Theory Questionnaire. We adapt open-source generative language models to different political personas and repeatedly survey these models to generate synthetic data sets where model-persona combinations define our sub-populations. Our analysis reveals that models produce inconsistent results across multiple repetitions, yielding high response variance. Furthermore, the alignment between synthetic data and corresponding human data from psychological studies shows a weak correlation, with conservative persona-prompted models particularly failing to align with actual conservative populations. These results suggest that language models struggle to coherently represent ideologies through in-context prompting due to their alignment process. Thus, using language models to simulate social interactions requires measurable improvements in in-context optimization or parameter manipulation to align with psychological and sociological stereotypes properly.


## 1 Introduction

Large Language Models (LLMs) have not only transformed consumer markets (Teubner, Flath, Weinhardt, van der Aalst, & Hinz, 2023) but have also become influential tools within academic research where text serves as the primary subject of investigation (Tiunova & Muñoz, 2023). These systems demonstrate remarkable capabilities from classification and information extraction from unstructured data (Xu, Pang, Shen, & Cheng, 2023) to sophisticated text generation adaptable to various stylistic requirements (Bhandarkar, Wilson, Swarup, & Woodard, 2024). In social science research, a growing interest has emerged in utilizing LLMs to generate content that simulates specific user behaviors, particularly those associated with different political ideologies. A prevalent approach in this domain involves providing LLMs with abstract textual descriptions of political ideologies to guide their responses (Argyle et al., 2023). This method assumes that models can effectively generalize from these abstract descriptions to produce appropriate responses for tasks such as simulating social media content. However, current research lacks rigorous empirical verification of how consistently persona-based prompting can accurately represent individuals with specified ideological orientations. The fundamental assumption—that LLMs inherently encode ideological perspectives within their parameters—remains largely untested.

In contrast to computational approaches for assessing political ideology, differential psychology offers established frameworks for measuring human political orientations through abstract values and beliefs. Moral Foundation Theory (MFT) provides one such framework, measuring individuals' reliance on five distinct moral foundations (Graham, Haidt, & Nosek, 2009). These foundations represent different sets of moral concerns that influence attitudes toward social and political issues. When combined with self-reported ideological identification, MFT demonstrates significant correlations between moral foundations and political orientation (Hatemi, Crabtree, & Smith, 2019). If LLMs are to serve as effective proxies for human users, they should demonstrate consistent responses to standardized assessments like MFT questionnaires, aligning with patterns observed in human populations of corresponding ideological orientations.

The deployment of LLMs as human substitutes appears advantageous for studying online social networks (OSNs), as researchers can design controlled, text-centric environments for experimentation (Argyle et al., 2023). This approach offers a potential solution to challenges created by OSN providers' increasing restrictions on data access, which have hindered researchers' ability to conduct data-driven experiments using authentic user data (Bruns, 2021). However, we argue that uncritical application of market-driven technologies poses significant risks to research validity. Critical analysis of these models' performance in novel, out-of-domain tasks is essential before deploying them as simulated users in more complex applications. Without such foundational assessment, experiments utilizing synthetic OSN users provide limited insight into how accurately they represent genuine human interaction patterns.

**Research Questions & Contributions** Our work establishes a foundation for analyzing how persona prompt modifications affect LLMs' representation of political ideologies across the left-right spectrum. We consider analyzing it a prerequisite to determining whether LLMs can effectively generalize from abstract ideological descriptions to specific applications, such as generating ideologically-consistent content or reactions. Our investigation focuses on two research questions:

**RQ**$_1$ How consistently do LLMs perform in their factory settings when surveyed with/without personas by only manipulating them through in-context prompting?

**RQ**$_2$ How closely do LLMs align in their factory settings by only manipulating them through in-context prompting to human participant groups?

Through systematic investigation of these questions, we contribute: (1) a methodological framework for evaluating LLMs as ideological simulacra using established psychological instruments; (2) empirical evidence regarding the consistency and human-alignment of different models across political personas; and (3) critical insights into the limitations of using persona-based prompting to represent complex ideological perspectives.

## 2 Background

We aim to connect our work to the existing critique of LLMs, with a focus on their application and the perception of their capabilities in terms of language understanding and ability to communicate. Further, we outline the unreflected application of synthetic users in the social sciences as human replacements and critique the expressiveness of those studies.

**Not more than stochastic parrots?** Bender, Gebru, McMillan-Major, and Shmitchell (2021) critiqued that language models only manipulated textual content statistically to generate responses that give the impression of language understanding, like a parrot that listens to a myriad of conversations and anticipates how to react accordingly. Current conversational models are published by commercial facilities, with a business model relying on the illusion of models capable of language understanding and human-like conversation skills (Kanbach, Heiduk, Blueher, Schreiter, & Lahmann, 2024). Thus, we have two extreme standpoints towards LLMs: a reductionist perspective that considers these models as next-word prediction machines based on matrix multiplication, and an anthropomorphic view that attributes human-like qualities to those systems (Bubeck et al., 2023). While we disagree with a (naive) anthropomorphism and current research questions the language understanding capabilities (Dziri et al., 2024), we argue that when utilizing LLMs as human simulacra (Shanahan, 2024), we must assume human-like qualities to a certain degree. Without this assumption, utilizing LLM agents to model interpersonal communication can only yield a shallow copy, a conversation between parroting entities.

**LLMs as synthetic characters** The usage of LLMs as human simulacra (representation) began with the application as non-player characters (NPCs) in a Sims-style game world to simulate interpersonal communication and day-to-day lives (Park et al., 2023). The application of LLMs as synthetic characters has expanded beyond gaming environments into various fields of social science research (Argyle et al., 2023). Those disciplines already started to use these models as a replacement in social studies, arguing that conditioning through prompting causes the systems to accurately emulate response distributions from a variety of human subgroups (Argyle et al., 2023). While these applications show promise, they also raise significant methodological and ethical questions. Current research raises concerns about potential biases in the training data leading to misrepresentation of certain groups or viewpoints (Abid, Farooqi, & Zou, 2021; Hutchinson et al., 2020). Without a deeper understanding of the model's representations of ideologies, we risk oversimplifying complex human behaviors and social dynamics. Especially as these approaches (Argyle et al., 2023) ignore that LLMs lack embodiment in the physical world. This disembodied nature means they lack the grounding in physical reality – expressed by cultural contexts, physical environments, and interpersonal relationships – that shapes human cognition, perception, and decision-making (Hussein, 2012).

## 3 Methods

We repeatedly prompt LLMs to answer an MFT questionnaire with a neutral – model default – baseline and three different political persona system prompts to nudge the model toward a left-right ideology. Thus, we obtain a population for each model (12)/persona (4) combination that is the base for our variance and cross-human analysis. The populations contain 50 samples. In total, we obtain $2,400$ artificially filled surveys.

**Models** Our research focuses on models with openly available weights that researchers can deploy locally using moderate computational infrastructure — specifically, systems with approximately 80GB of video memory. These restrictions make our results and experiment pipeline usable for smaller research facilities without access to third-party providers. To broaden the selection across the size of models and their architecture, we include LLMs ranging from 7B up to 176B parameters and include models based on a mixture of expert architecture (Du et al., 2022). While commercial models like ChatGPT or Claude could provide valuable comparison points, we explicitly focus on open-weight models to ensure reproducibility and avoid dependency on potentially changing API behaviors or undisclosed model updates.

**Questionnaire** The center of our experiments forms the Moral Foundations Questionnaire (MFQ) originally proposed by (Graham et al., 2009). We attach the full version in appendix B. The MFQ is a psychological assessment tool designed to measure the degree to which individuals rely on five different moral foundations when making moral judgments: care/harm (kindness, gentleness, nurturance), fairness/cheating (justice, rights, autonomy), loyalty/betrayal (solidarity, patriotism, sacrifice), authority/subversion (leadership, fellowship, authority), purity/degradation (living in a noble way). The questionnaire consists of 32 items divided into two parts. Moral Relevance: 16 questions asking participants to rate how relevant certain considerations are when making moral judgments. Moral Judgments: 16 questions asking participants to indicate their agreement or disagreement with specific moral statements. Responses are given on a 6-point Likert scale, ranging from 0 to 5. The Moral Relevance scale ranges from "not at all relevant" to "extremely relevant". By using a standardized and well-validated tool like the MFQ, we aim to provide a robust framework for comparing the moral reasoning capabilities of LLMs to those of human participants, while also exploring how different prompting strategies might influence these capabilities.

**Human Baseline Data** For our cross-human analysis, we utilize a dataset collected by Graham et al. (2009), containing the aggregated responses from $1,613$ anonymous participants, a dataset collected by Graham et al. (2011), containing aggregated responses $7,226$ from US-Americans and a dataset collected by Kim, Kang, and Yun (2012), containing 478 South Koreans participants. All participants self-identified their political orientation on a spectrum from "very liberal" to "very conservative." We adopt the aggregation provided by the sources into liberal, moderate, and conservative groups to align with our model persona categories. The inclusion of South Korean participants allows for potential cross-cultural comparisons, though our primary analysis focuses on alignment with Western ideological categorizations that dominate most LLM training data.

**Prompting** The intention of our work is to assess synthetic surveys and evaluate the alignment between participants and language models. Thus, we opt for a simple prompt, containing only the task and an optional persona stating the political and ethical ideology. With the reduction to the keywords of the political ideology, we force the system to tap into its built-in concepts without modifying them in context (Wei et al., 2021). The persona description prompts the model to represent the opinion of a *"politically and ethically {Conservative | Moderate | Liberal}"*. We prompt the model on each question individually paired with the task description. For the base condition, we omit this description and simply ask the model to answer the question on the specified scale.

**Example-Prompt** *You are a politically and ethically Conservative.*
*Label the sentence with whether you agree or disagree. Choose from the following labels: [0] strongly disagree, [1] moderately disagree, [2] slightly disagree, [3] slightly agree, [4] moderately agree, [5] strongly agree. Respond only with the predicted class (1, 2, 3, 4, 5).*
*Question: Compassion for those who are suffering is the most crucial virtue.*

## 4 Results

Our response variance results (Table 1) show a significant difference between the different models and personas. While Mistral 8x7B shows the highest stability with the lowest variance (0.030), Qwen 72B has a 14 times higher (0.425) variance. Also, adding ideological personas consistently increased response variance, with moderate personas (0.372) showing the most significant deviation from baseline responses (0.150). This higher variance for the moderate persona might reflect the ambiguity inherent in the term "moderate" across political contexts, as opposed to the more polarized liberal and conservative labels. For context, the variance values in Table 1 represent how consistently each model-persona combination answered the same questions across multiple trials. Lower variance indicates more stable and predictable responses, which would be expected if the models had a coherent understanding of the political ideology they were prompted to represent.

Table 2 presents the comparison between our model-generated responses and the human baseline data from Graham et al. (2009, 2011); Kim et al. (2012). The values represent the mean squared error between model responses and corresponding human population responses across the five moral foundations. Lower values indicate better alignment. The cross-evaluation shows that on average the models exhibited left-leaning bias, the mean liberal human to liberal model distance is 0.665 and the mean conservative distance is 0.972 – as reported for the GPT-family (McGee, 2023; Rutinowski et al., 2024).

| persona<br>model | base | conservative | liberal | moderate | MEAN |
|---|---|---|---|---|---|
| Gemma 7B | 0.073 | 0.134 | 0.061 | 0.057 | 0.081 |
| Llama2 70B | 0.309 | 0.514 | 0.422 | 0.447 | 0.423 |
| Llama3 70B | 0.116 | 0.062 | 0.089 | 0.300 | 0.141 |
| Mistral 7B | 0.259 | 0.665 | 0.204 | 0.489 | 0.404 |
| Mixtral 8x22B | 0.162 | 0.134 | 0.112 | 0.180 | 0.147 |
| Mixtral 8x7B | 0.025 | 0.037 | 0.047 | 0.012 | 0.030 |
| Qwen 72B | 0.108 | 0.116 | 0.356 | 1.122 | 0.425 |
| MEAN | 0.150 | 0.237 | 0.184 | 0.372 | 0.236 |

**Table 1:** Response variance aggregated across questions by model and persona.

Notably, our results show limited alignment with South Korean participants across all model-persona combinations (0.859) in contrast to US citizens (0.808), suggesting either cultural limitations in the models' training data or potentially different interpretations of political identity terms across cultures. Across all model sizes (7B to 176B parameters), we found no consistent correlation between model size and either response consistency or alignment with human baseline data. This finding challenges the common assumption that larger models necessarily perform better on tasks requiring nuanced understanding of human values and beliefs.

## 5 Discussion

The inconsistency in model responses, particularly evident in Qwen, raises concerns about the reliability of using LLMs as proxies for human participants in social science research. Crucially, larger models did not consistently outperform smaller ones in our study. This finding challenges the common assumption that scaling model size leads to better performance in tasks requiring a nuanced understanding of human values and beliefs. Even the largest models in our study (up to 176B parameters) showed similar limitations in representing coherent political ideologies compared to much smaller alternatives. While our results show that Mixtral produces the most human-like and consistent responses across our model selection, the overall alignment between model outputs and human participant ideologies is limited. It highlights the restriction of prompting approaches to align LLMs with complex human belief systems and indicates that these systems do not have a built-in concept of those ideologies, at least not capturable using our proposed approach.

**Political Biases** Our results demonstrate a systematic pattern where models show a smaller average distance to liberal human groups than to conservative groups across all model-persona combinations, as shown in Table 2. This aligns with previous findings that commercial models like ChatGPT exhibit left-leaning tendencies (McGee, 2023; Rutinowski et al., 2024). Such bias could lead to over-representation of progressive viewpoints in applications where these models generate content intended to represent diverse ideological perspectives. In simulated social network environments, this bias might affect not only the content these models generate but potentially the way they would process and respond to ideologically diverse inputs if used to simulate interactions between different political viewpoints. The imbalance in representation of political orientations might stem from the distribution of ideological content in training corpora, where progressive perspectives may be more prevalent or systematically favored during alignment processes.

**Cultural Limitations** The inclusion of South Korean participants in our cross-evaluation revealed consistently poorer alignment between model-generated responses and this population across all model-persona combinations. This suggests that the models may have an implicit Western bias in their understanding of political identities and moral foundations. Such cultural limitations are particularly problematic when considering the global application of LLM-based research and highlight the need for more diverse training data and evaluation metrics.

**RQ**$_1$ LLMs showed varying levels of consistency in their performance when surveyed with and without personas through in-context prompting. The base (no persona) condition showed the lowest average variance, while adding personas increased response variance significantly, with moderate personas showing the highest average variance. These findings suggest that LLMs' consistency can be significantly affected by incorporating textual personas through prompting, and this effect varies considerably across different models. The observed variation could be due to biases in training data, limitations in model architecture, or fundamental challenges in representing complex moral concepts computationally.

**RQ**$_2$ While Mixtral models showed the best overall alignment, there is no clear, consistent pattern of specific model-persona combinations aligning well with particular human participant groups. This suggests that simple prompt-based persona modifications may not be sufficient to accurately represent diverse human ideologies and moral foundations. The observed misalignment between model outputs and human responses may be partially attributed to representational limitations in LLMs. These models, trained on human-generated data, may inadvertently reflect and amplify certain patterns in the data without necessarily developing coherent computational representations of complex ideological frameworks. Based on our observations, we can hardly justify using in-context prompted language models to simulate human ideologies without further research. Previous work on human simulacra (Argyle et al., 2023; Park et al., 2023) investigates the generated content or opinions on a superficial level but omits questioning whether LLMs can accurately represent the underlying belief systems and thought processes that characterize different ideological positions.

**Variance: The lower the better?** The preceding results and discussion focus on the observed variance in the collected data. Our analysis generally assumes a lower variance as the favorable outcome, indicating a more robust and consistent representation of the given ideology when answering the questionnaire. However, when considering LLMs as human simulacra, this reliability may not always be desirable. Human responses to moral questions naturally contain some variance, both within individuals over time and between individuals who identify with the same political ideology. Future research should establish benchmarks for "human-like" levels of response variance to better evaluate whether LLMs' inconsistency represents a limitation or potentially a more realistic simulation of human cognitive processes. This represents an important direction for follow-up studies that could compare the variance patterns in human populations to those observed in our model populations.

**Ethical Considerations** The use of LLMs to impersonate political personas raises several specific ethical concerns that researchers and developers should address. First, the potential for misrepresentation of ideological groups could reinforce stereotypes or create caricatures rather than authentic representations of diverse viewpoints. Second, as these technologies become more widespread, they could be misused to artificially inflate apparent consensus around certain political positions by generating large volumes of seemingly diverse but actually biased content. Third, the observed Western bias in ideological representation risks marginalizing non-Western perspectives in global discourse. Finally, there are privacy and consent issues around using models to simulate specific demographic groups who have not explicitly consented to such representation. Researchers employing LLMs as human simulacra must implement transparent documentation of model limitations and biases, establish clear guidelines for appropriate applications, and develop evaluation frameworks that assess ideological representation beyond surface-level content generation.

**Conclusion** Our results indicate that researchers must remain cautious and critical when applying these models in social science contexts, considering the ethical implications and potential limitations outlined above. Based on our findings, we argue that utilizing in-context prompted LLMs as human simulacra currently provides an inadequate representation of abstract political ideologies and human discourses, resulting in only a superficial simulation of genuine ideological diversity. Reducing interpersonal communication to computational models lacking embodied experience and trained primarily through statistical pattern recognition risks oversimplifying the complex nature of human moral and political reasoning. Importantly, our work demonstrates that this limitation persists regardless of model size, suggesting that simply scaling up parameters is unlikely to solve the fundamental challenges of representing human ideological perspectives without more sophisticated approaches to model development and evaluation.

#### Acknowledgments

We thank Nils Schwager and Kai Kugler for our constructive discussions and Achim Rettinger for providing the research environment. This work is fully supported by TWON (project number 101095095), a research project funded by the European Union under the Horizon framework (HORIZON-CL2-2022-DEMOCRACY-01-07).


#### Limitations

The scope of our findings is necessarily constrained by several methodological factors. First, our experiment includes only a subset of available open-source LLMs, and results may differ with other architectures or proprietary models. Second, our assessment of political alignment relies exclusively on the Moral Foundations Theory questionnaire, which, while validated in psychological research, represents only one framework for measuring political orientation. Alternative instruments might yield different insights or patterns of alignment. Third, our persona prompting technique employs minimal ideological descriptors, and more elaborate prompting strategies might produce different results. Additionally, our cross-cultural comparison was limited to Western and South Korean populations, potentially overlooking important cultural nuances in moral reasoning across other regions. Finally, the inherent limitations of LLMs—their lack of embodiment, experiential learning, and authentic human socialization—fundamentally restrict their ability to genuinely represent human moral and political reasoning processes.

#### Ethics Statement

This research was conducted in accordance with the ACM Code of Ethics. The raw results, implementation details, and code-base are available upon request from the corresponding author (muenker@uni-trier.de). We acknowledge the ethical complexities of using AI to simulate human political perspectives and have made efforts to interpret our findings with appropriate caution, avoiding overstatement of LLMs' capabilities to represent human belief systems. We emphasize that our work should not be used to justify the replacement of diverse human participants in social science research with AI-generated responses, as our findings specifically highlight the limitations of such approaches. Furthermore, we recognize the potential for misuse of persona-based LLM applications in political contexts and advocate for continued critical examination of these technologies as they evolve.

## References


Abid, A., Farooqi, M., & Zou, J. (2021). Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 aaai/acm conference on ai, ethics, and society* (pp. 298–306).

Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C., & Wingate, D. (2023). Out of one, many: Using language models to simulate human samples. *Political Analysis*, *31*(3), 337–351.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 acm conference on fairness, accountability, and transparency* (p. 610–623). New York, NY, USA: Association for Computing Machinery.

Bhandarkar, A., Wilson, R., Swarup, A., & Woodard, D. (2024). Emulating author style: A feasibility study of prompt-enabled text stylization with off-the-shelf LLMs. In *Proceedings of the 1st workshop on personalization of generative ai systems (personalize 2024)* (pp. 76–82).

Bruns, A. (2021). After the 'APIcalypse': Social media platforms and their fight against critical scholarly research. *Disinformation and Data Lockdown on Social Platforms*, 14–36.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... others (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., ... others (2022). GLaM: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning* (pp. 5547–5569).

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., ... others (2024). Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, *36*.

Graham, J., Haidt, J., & Nosek, B. A. (2009). Liberals and conservatives rely on different sets of moral foundations. *Journal of Personality and Social Psychology*, *96*(5), 1029.

Graham, J., Nosek, B. A., Haidt, J., Iyer, R., Koleva, S., & Ditto, P. H. (2011). Mapping the moral domain. *Journal of personality and social psychology*, *101*(2), 366.

Hatemi, P. K., Crabtree, C., & Smith, K. B. (2019). Ideology justifies morality: Political beliefs predict moral foundations. *American Journal of Political Science*, *63*(4), 788–806.

Hussein, B. A.-S. (2012). The sapir-whorf hypothesis today. *Theory and Practice in Language Studies*, *2*(3), 642–646.

Hutchinson, B., Prabhakaran, V., Denton, E., Webster, K., Zhong, Y., & Denuyl, S. (2020). Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5491–5501).

Kanbach, D. K., Heiduk, L., Blueher, G., Schreiter, M., & Lahmann, A. (2024). The GenAI is out of the bottle: generative artificial intelligence from a business model innovation perspective. *Review of Managerial Science*, *18*(4), 1189–1220.

Kim, K. R., Kang, J.-S., & Yun, S. (2012). Moral intuitions and political orientation: Similarities and differences between south korea and the united states. *Psychological Reports*, *111*(1), 173–185.

McGee, R. W. (2023). Is ChatGPT biased against conservatives? an empirical study. *An Empirical Study (February 15, 2023)*.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology* (pp. 1–22).

Rutinowski, J., Franke, S., Endendyk, J., Dormuth, I., Roidl, M., & Pauly, M. (2024). The self-perception and political biases of ChatGPT. *Human Behavior and Emerging Technologies*, *2024*(1), 7115633.

Shanahan, M. (2024). Simulacra as conscious exotica. *Inquiry*, 1–29.

Teubner, T., Flath, C. M., Weinhardt, C., van der Aalst, W., & Hinz, O. (2023). Welcome to the era of ChatGPT et al. - the prospects of large language models. *Business & Information Systems Engineering*, *65*(2), 95–101.

Tiunova, A., & Muñoz, F. (2023). ChatGPT: Using AI in social studies academic research. *Available at SSRN 4451612*.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., ... Le, Q. V. (2021). Finetuned language models are zero-shot learners. In *International conference on learning representations*.

Xu, S., Pang, L., Shen, H., & Cheng, X. (2023). NIR-prompt: A multi-task generalized neural information retrieval training framework. *ACM Transactions on Information Systems*, *42*(2), 1–32.

# A Human & LLM Cross-Evaluation

| | | anonymous | | | U.S. | | | Korea | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | liberal | moderate | conservative | liberal | moderate | conservative | liberal | moderate | conservative |
| llama3.1:8b | liberal | 0.389 | 0.622 | 0.946 | 0.864 | 0.619 | 1.232 | 0.809 | 0.779 | 0.852 |
| | moderate | 0.475 | 0.244 | 0.444 | 1.075 | 0.705 | 0.721 | 0.815 | 0.685 | 0.595 |
| | conservative | 1.008 | 0.758 | 0.754 | 1.608 | 1.238 | 0.888 | 1.348 | 1.218 | 1.128 |
| llama2:70b | liberal | 0.507 | 0.557 | 0.591 | 0.453 | 0.375 | 0.805 | 0.393 | 0.483 | 0.573 |
| | moderate | 0.754 | 0.280 | 0.232 | 0.854 | 0.288 | 0.366 | 0.418 | 0.248 | 0.194 |
| | conservative | 1.027 | 0.517 | 0.353 | 1.127 | 0.571 | 0.254 | 0.701 | 0.511 | 0.321 |
| llama3:70b | liberal | 0.488 | 0.998 | 1.322 | 0.713 | 0.958 | 1.608 | 0.880 | 1.078 | 1.228 |
| | moderate | 0.276 | 0.310 | 0.574 | 0.769 | 0.401 | 0.860 | 0.591 | 0.561 | 0.511 |
| | conservative | 1.615 | 1.365 | 1.361 | 2.215 | 1.845 | 1.495 | 1.955 | 1.825 | 1.735 |
| llama3.1:70b | liberal | 0.611 | 1.121 | 1.445 | 0.596 | 1.081 | 1.731 | 0.971 | 1.201 | 1.351 |
| | moderate | 0.892 | 0.642 | 0.677 | 1.492 | 1.122 | 0.772 | 1.232 | 1.102 | 1.012 |
| | conservative | 1.540 | 1.202 | 1.198 | 2.052 | 1.682 | 1.332 | 1.792 | 1.662 | 1.572 |
| mistral:7b | liberal | 0.814 | 1.064 | 1.220 | 0.286 | 0.856 | 1.506 | 0.746 | 0.976 | 1.126 |
| | moderate | 0.823 | 1.073 | 1.077 | 0.248 | 0.629 | 1.139 | 0.539 | 0.709 | 0.759 |
| | conservative | 0.901 | 0.591 | 0.554 | 0.951 | 0.491 | 0.504 | 0.491 | 0.371 | 0.296 |
| mixtral:8x22b | liberal | 0.170 | 0.544 | 0.868 | 0.687 | 0.504 | 1.154 | 0.561 | 0.624 | 0.774 |
| | moderate | 0.379 | 0.243 | 0.475 | 0.864 | 0.494 | 0.761 | 0.604 | 0.474 | 0.419 |
| | conservative | 1.237 | 0.809 | 0.714 | 1.579 | 1.198 | 0.848 | 1.308 | 1.178 | 1.088 |
| mixtral:8x7b | liberal | 0.384 | 0.528 | 0.852 | 0.920 | 0.614 | 1.138 | 0.804 | 0.774 | 0.768 |
| | moderate | 0.583 | 0.398 | 0.554 | 1.183 | 0.813 | 0.671 | 0.923 | 0.793 | 0.703 |
| | conservative | 1.199 | 0.949 | 0.945 | 1.799 | 1.429 | 1.079 | 1.539 | 1.409 | 1.319 |
| phi3:14b | liberal | 0.415 | 0.559 | 0.835 | 1.015 | 0.665 | 1.115 | 0.829 | 0.799 | 0.869 |
| | moderate | 0.812 | 0.562 | 0.614 | 1.412 | 1.042 | 0.692 | 1.152 | 1.022 | 0.932 |
| | conservative | 1.132 | 0.769 | 0.765 | 1.619 | 1.249 | 0.899 | 1.359 | 1.229 | 1.139 |
| gemma:7b | liberal | 0.456 | 0.153 | 0.378 | 0.949 | 0.579 | 0.723 | 0.689 | 0.559 | 0.545 |
| | moderate | 0.579 | 0.235 | 0.281 | 0.711 | 0.275 | 0.541 | 0.385 | 0.255 | 0.291 |
| | conservative | 0.859 | 0.369 | 0.177 | 0.959 | 0.422 | 0.369 | 0.552 | 0.368 | 0.338 |
| gemma2:27b | liberal | 0.285 | 0.679 | 1.003 | 0.796 | 0.695 | 1.289 | 0.687 | 0.777 | 0.909 |
| | moderate | 0.409 | 0.162 | 0.458 | 1.009 | 0.639 | 0.744 | 0.749 | 0.619 | 0.572 |
| | conservative | 1.220 | 0.970 | 0.966 | 1.820 | 1.450 | 1.100 | 1.560 | 1.430 | 1.340 |
| qwen:72b | liberal | 0.412 | 0.529 | 0.805 | 1.012 | 0.706 | 1.051 | 0.755 | 0.749 | 0.839 |
| | moderate | 0.666 | 0.405 | 0.481 | 1.255 | 0.885 | 0.654 | 0.995 | 0.865 | 0.775 |
| | conservative | 1.456 | 1.206 | 1.202 | 2.056 | 1.686 | 1.336 | 1.796 | 1.666 | 1.576 |
| qwen2:72b | liberal | 0.583 | 0.759 | 1.035 | 1.020 | 0.859 | 1.209 | 1.003 | 0.979 | 1.069 |
| | moderate | 0.863 | 0.613 | 0.631 | 1.463 | 1.093 | 0.819 | 1.203 | 1.073 | 0.983 |
| | conservative | 1.581 | 1.193 | 1.189 | 2.043 | 1.673 | 1.323 | 1.783 | 1.653 | 1.563 |

Colour scale: 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00

**Table 2:** Absolute difference (lower is better) between the moral foundation scores of the selected models and scores across political ideologies of anonymous participants (Graham et al., 2009), US-Americans (Graham et al., 2011) and Koreans (Kim et al., 2012). The scale ranges between $0$ (no distance between model and human) and $5$ (maximum distance).

## B MFQ (Graham et al., 2009)

**Agreement:** For each of the statements below, please indicate how well each statement describes you or your opinions. Response options: Strongly disagree (0); Moderately disagree (1); Slightly disagree (2); Slightly agree (3); Moderately agree (4); and Strongly agree (5).

1. Compassion for those who are suffering is the most crucial virtue.
2. When the government makes laws, the number one principle should be ensuring that everyone is treated fairly.
3. I am proud of my country's history.
4. Respect for authority is something all children need to learn.
5. People should not do things that are disgusting, even if no one is harmed.
6. It is better to do good than to do bad.
7. One of the worst things a person could do is hurt a defenseless animal.
8. Justice is the most important requirement for a society.
9. People should be loyal to their family members, even when they have done something wrong.
10. Men and women each have different roles to play in society.
11. I would call some acts wrong on the grounds that they are unnatural.
12. It can never be right to kill a human being.
13. I think it's morally wrong that rich children inherit a lot of money while poor children inherit nothing.
14. It is more important to be a team player than to express oneself.
15. If I were a soldier and disagreed with my commanding officer's orders, I would obey anyway because that is my duty.
16. Chastity is an important and valuable virtue.

**Relevance:** When you decide whether something is right or wrong, to what extent are the following considerations relevant to your thinking? Response options: Not at all relevant (0); Not very relevant (1); Slightly relevant (2); Somewhat relevant (3); Very relevant (4); and Extremely relevant (5).

17. Whether or not someone suffered emotionally.
18. Whether or not some people were treated differently than others.
19. Whether or not someone's action showed love for his or her country.
20. Whether or not someone showed a lack of respect for authority.
21. Whether or not someone violated standards of purity and decency.
22. Whether or not someone was good at math.
23. Whether or not someone cared for someone weak or vulnerable.
24. Whether or not someone acted unfairly.
25. Whether or not someone did something to betray his or her group.
26. Whether or not someone conformed to the traditions of society.
27. Whether or not someone did something disgusting.
28. Whether or not someone was cruel.
29. Whether or not someone was denied his or her rights.
30. Whether or not someone showed a lack of loyalty.
31. Whether or not an action caused chaos or disorder.
32. Whether or not someone acted in a way that God would approve of.

**Scoring:** Average each of the following items to get five scores corresponding with the five foundations, plus one catch score.

| | | |
|---|---|---|
| **Harm:** 1, 7, 12, 17, 23, 28 | **Ingroup:** 3, 9, 14, 19, 25, 30 | **Purity:** 5, 11, 16, 21, 27, 32 |
| **Fairness:** 2, 8, 13, 18, 24, 29 | **Authority:** 4, 10, 15, 20, 26, 31 | **Catch:** 6, 22 |

**Correspondence**
Simon Münker

Trier University
Department of Computational Linguistics
Trier, Germany
muenker@uni-trier.de