StarSEM 2025

The 14th Joint Conference on Lexical and Computational Semantics

Proceedings of the Conference (*SEM 2025)

November 8-9, 2025

The StarSEM organizers gratefully acknowledge the support from the following sponsors.

Sponsored by

©2025 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
`acl@aclweb.org`

ISBN 979-8-89176-340-1

Preface by the Conference Organizers

We are excited to welcome you to *SEM 2025, the 14th Joint Conference on Lexical and Computational Semantics! We are pleased to present this volume containing the accepted long and short papers. *SEM 2025 is being held from November 8 to 9, 2025, in Suzhou, China, co-located with EMNLP 2025.

Since its first edition in 2012, *SEM has become a major venue to present recent advances in all areas of lexical and computational semantics, including semantic representations, theoretical semantics, multilingual semantics, and others. *SEM is sponsored by SIGLEX, the ACL Special Interest Group on the Lexicon.

*SEM 2025 accepted both papers submitted directly to *SEM and those already reviewed through ARR (ACL Rolling Review). We received submissions in 4 tracks:

- Empirical and data-driven approaches
- Theoretical aspects of computational semantics
- Applications or tools
- Resources and evaluation

We compiled an exciting and wide-ranging program, accepting a total of 35 papers (26 long papers and 9 short papers). The submitted papers were carefully evaluated by a program committee led by 10 area chairs, who coordinated a large team of reviewers. The reviews were almost all of very high-quality, and for that we are extremely grateful! Area chairs then added meta-reviews to explain their accept/reject decisions. The final selection was made by the program co-chairs after a careful check of the reviews, meta-reviews, and discussions with the area chairs.

We are also very excited to have three excellent keynote speakers: Yue Zhang from Westlake University, Thamar Solorio from Mohamed bin Zayed University of Artificial Intelligence (MBZUAI) and Yuki Arase from the Tokyo Institute of Technology.

We are honored to serve as the organizing committee for *SEM 2025, and we absolutely could not have made this happen without a huge amount of help. First, tremendous thanks to all area chairs and reviewers for their invaluable help in selecting the program, for their engagement in thoughtful discussions, and for providing valuable feedback to authors. Second, thanks to our Publicity Chair Kemal Kurniawan (University of Melbourne) for taking care of the website and social media updates. Next, thanks to our Publication Chairs Danilo Croce (University of Rome) and Milad Alshomary (Columbia University) for putting together the proceedings, and to the EMNLP 2025 workshop organizers for help and support with all organizational aspects of the conference. Finally, thank you to the authors and presenters for making *SEM 2025 such an engaging and exciting event! We hope that you will find the content of these proceedings as engaging as we do, and we hope to see you at future iterations of *SEM!

Lea Frermann and Mark Stevenson, co-Program Chairs
Vered Shwartz, General Chair

Organizing Committee

General Chair

Vered Shwartz, University of British Columbia

Program Chairs

Lea Frermann, University of Melbourne
Mark Stevenson, University of Sheffield

Publication Chairs

Danilo Croce, University of Rome Tor Vergata
Milad Alshomary, Columbia University

Publicity Chair

Kemal Kurniawan, University of Melbourne

Program Committee

Area Chairs

Steven Bethard, University of Arizona
Lucia Donatelli, Vrije Universiteit Amsterdam
Fajri Koto, Mohamed bin Zayed University of Artificial Intelligence
Bridget McInnes, Virginia Commonwealth University
Siyao Peng, Ludwig-Maximilians-Universität München
Jakob Prange, University of Augsburg
Abhilasha Ravichander, University of Washington and Allen Institute for AI
Harish Tayyar Madabushi, University of Bath
Peter Vickers, Northeastern University
Shira Wein, Amherst College

Program Committee

Omri Abend, Hebrew University of Jerusalem
Ameeta Agrawal, Portland State University
Yuki Arase, Tokyo Institute of Technology
Naomi Baes, University of Melbourne
Timothy Baldwin, Mohamed bin Zayed University of Artificial Intelligence and The University of Melbourne
Mohamad Hardyman Barawi, Universiti Malaysia Sarawak
Gábor Berend, University of Szeged
Abhidip Bhattacharyya, University of Massachusetts at Amherst
Eduardo Blanco, University of Arizona
Michael Bloodgood, The College of New Jersey
Joanne Boisson, Cardiff University
Davide Buscaldi, Ecole polytechnique and Université Paris 13
Ming-Bin Chen, University of Melbourne
Mukund Choudhary, Mohamed bin Zayed University of Artificial Intelligence
Gabriella Chronis, University of Texas at Austin
Paul Cook, University of New Brunswick
Robin Cooper, University of Gothenburg
Bonaventura Coppola, University of Trento
Xingyu Deng, University of Sheffield
Jakub Dotlacil, Utrecht University
Bilal Elbouardi, Mohamed bin Zayed University of Artificial Intelligence
Katrin Erk, University of Texas, Austin
Francis Ferraro, University of Maryland, Baltimore County
Aaron Fletcher, University of Sheffield
Colin Gordon, Drexel University
Venkata Govindarajan, Ithaca College
Alvin Grissom Ii, Haverford College
Udo Hahn, Friedrich-Schiller-Universität Jena
Yoshihiko Hayashi, Waseda University
Pingjun Hong, University of Vienna
Antonio Jimeno Yepes, Unstructured Technologies
Muhammad Kautsar, Mohamed bin Zayed University of Artificial Intelligence

Halil Kilicoglu, University of Illinois at Urbana-Champaign
Alexander Koller, Saarland University
Grzegorz Kondrak, University of Alberta
Andrey Kutuzov, University of Oslo
Kenneth Lai, Brandeis University and Mass General Brigham
Els Lefever, Ghent University
Xixian Liao, Barcelona Supercomputing Center
Chunhua Liu, University of Melbourne
Aso Mahmudi, University of Melbourne
Eugenio Martínez-Cámara, Universidad de Jaén
Yisong Miao, National University of Singapore
Timothee Mickus, University of Helsinki
Koji Mineshima, Keio University
Amita Misra, Amazon
Lawrence Moss, Indiana University at Bloomington
Nurdaulet Mukhituly, Mohamed bin Zayed University of Artificial Intelligence
Nona Naderi, Université Paris-Saclay
Danial Namazifard, University of Tehran
Christina Niklaus, Universität St. Gallen
Amy Olex, Virginia Commonwealth University
Yulia Otmakhova, University of Melbourne
Emerson Paraiso, Pontifícia Universidade Católica do Paraná
Naiara Perez, University of the Basque Country (UPV/EHU)
Francesco Periti, KU Leuven
Massimo Poesio, Utrecht University and Queen Mary, University of London
Giulia Rambelli, University of Bologna
Leonardo Ranaldi, University of Edinburgh
Kyle Richardson, Allen Institute for Artificial Intelligence
Juan Rodriguez, University of Texas at Austin
Mehrnoosh Sadrzadeh, University College London
Steven Schockaert, Cardiff University
Sabine Schulte Im Walde, University of Stuttgart
Melanie Siegel, Darmstadt University of Applied Sciences
Elior Sulem, Ben-Gurion University of the Negev
Andon Tchechmedjiev, IMT Mines Alès
Junior Tonga, Mohamed bin Zayed University of Artificial Intelligence
Thinh Truong, University of Melbourne
L. Alfonso Ureña, Universidad de Jaén
Sowmya Vajjala, National Research Council Canada
Rossella Varvara, University of Turin
Eva Vecchi, University of Stuttgart
Serena Villata, CNRS
Tak-Lam Wong, Douglas College
Huiling You, University of Oslo
Shijia Zhou, Ludwig-Maximilians-Universität München

Table of Contents

ChengyuSTS: An Intrinsic Perspective on Mandarin Idiom Representation
Le Qiu, Emmanuele Chersoni and Aline Villavicencio 1

Bridging Information Gaps with Comprehensive Answers: Improving the Diversity and Informativeness of Follow-Up Questions
Zhe Liu, Taekyu Kang, Haoyu Wang, Seyed Hossein Alavi and Vered Shwartz 13

Injecting Frame Semantics into Large Language Models via Prompt-Based Fine-Tuning
Shahid Iqbal Rai, Danilo Croce and Roberto Basili 31

From Complex Word Identification to Substitution: Instruction-Tuned Language Models for Lexical Simplification
Tonghui Han, Xinru Zhang, Yaxin Bi, Maurice D. Mulvenna and Dongqiang Yang 48

Semantic Prosody in Machine Translation: the English-Chinese Case of Passive Structures
Xinyue Ma, Pol Pastells, Mariona Taulé Delor and Mireia Farrús 59

Semantic Differentiation in Speech Emotion Recognition: Insights from Descriptive and Expressive Speech Roles
Rongchen Guo, Vincent Francoeur, Isar Nejadgholi, Sylvain Gagnon and Miodrag Bolic 70

Generalizability of Media Frames: Corpus creation and analysis across countries
Agnese Daffara, Sourabh Dattawad, Sebastian Padó and Tanise Ceron 83

Cross-Lingual Extractive Question Answering with Unanswerable Questions
Yuval Gorodissky, Elior Sulem and Dan Roth 100

Evaluating Compositional Generalisation in VLMs and Diffusion Models
Beth Pearson, Bilal Boulbarss, Michael Wray and Martha Lewis 122

On the Distinctive Co-occurrence Characteristics of Antonymy
Zhihan Cao, Hiroaki Yamada and Takenobu Tokunaga 134

Evaluating Textual and Visual Semantic Neighborhoods of Abstract and Concrete Concepts
Sven Naber, Diego Frassinelli and Sabine Schulte Im Walde 142

AmbiStory: A Challenging Dataset of Lexically Ambiguous Short Stories
Janosch Gehring and Michael Roth 152

WiC Evaluation in Galician and Spanish: Effects of Dataset Quality and Composition
Marta Vázquez Abuín and Marcos Garcia 172

Math Natural Language Inference: this should be easy!
Valeria de Paiva, Qiyue Gao, Hai Hu, Pavel Kovalev, Yikang Liu, Lawrence S. Moss and Zhiheng Qian 179

All Entities are Not Created Equal: Examining the Long Tail for Ultra-Fine Entity Typing
Advait Deshmukh, Ashwin Umadi, Dananjay Srinivas and Maria Leonor Pacheco 189

When Does Meaning Backfire? Investigating the Role of AMRs in NLI
Junghyun Min, Xiulin Yang and Shira Wein 202

Explanations explained. Influence of Free-text Explanations on LLMs and the Role of Implicit Knowledge
Andrea Zaninello, Roberto Dessi, Malvina Nissim and Bernardo Magnini 212

Latent Traits and Cross-Task Transfer: Deconstructing Dataset Interactions in LLM Fine-tuning
Shambhavi Krishna, Haw-Shiuan Chang and Taesung Lee 225

LLMs as annotators of argumentation
Anna Lindahl 242

If We May De-Presuppose: Robustly Verifying Claims through Presupposition-Free Question Decomposition
Shubhashis Roy Dipta and Francis Ferraro 253

Modeling Language Learning in Corrective Feedback Interactions
Juan Luis Castro-Garcia and Parisa Kordjamshidi 267

Relation-Aware Prompting Makes Large Language Models Effective Zero-shot Relation Extractors
Mahdi Rahimi, Razvan-Gabriel Dumitru and Mihai Surdeanu 280

Enhancing Readability-Controlled Text Modification with Readability Assessment and Target Span Prediction
Liu Fengkai and John Sie Yuen Lee 293

TAG–EQA: Text–And–Graph for Event Question Answering via Structured Prompting Strategies
Maithili Sanjay Kadam and Francis Ferraro 304

DisCoCLIP: A Distributional Compositional Tensor Network Encoder for Vision-Language Understanding
Kin Ian Lo, Hala Hawashin, Mina Abbaszadeh, Tilen Gaetano Limbäck-Stokin, Hadi Wazni and Mehrnoosh Sadrzadeh 316

HSGM: Hierarchical Segment-Graph Memory for Scalable Long-Text Semantics
Dong Liu and Yanxuan Yu 328

Knowledge Editing Induces Underconfidence in Language Models
Ryo Hasegawa, Yusuke Sakai, Hidetaka Kamigaito and Taro Watanabe 338

How Do Large Language Models Evaluate Lexical Complexity?
Abdelhak Kelious, Mathieu Constant and Christophe Coeur 348

SAG: Enhancing Domain-Specific Information Retrieval with Semantic-Augmented Graphs
Carol-Luca Gasan and Vasile Pais 362

Cross-Domain Persuasion Detection with Argumentative Features
Bagyasree Sudharsan and Maria Leonor Pacheco 372

Hallucinated Span Detection with Multi-View Attention Features
Yuya Ogasa and Yuki Arase 381

AdvERSEM: Adversarial Robustness Testing and Training of LLM-based Groundedness Evaluators via Semantic Structure Manipulation
Kaustubh Dhole, Ramraj Chandradevan and Eugene Agichtein 395

Connecting Concept Layers and Rationales to Enhance Language Model Interpretability
Thomas Bailleux, Tanmoy Mukherjee, Pierre Marquis and Zied Bouraoui 409

Towards Evaluation of Language Models with Skill Dimensions: A Case Study on Narrative Question Answering
Emil Kalbaliyev and Kairit Sirts 430

Potentially Problematic Word Usages and How to Detect Them: A Survey
Aina Garí Soler, Matthieu Labeau and Chloé Clavel 441

ChengyuSTS: An Intrinsic Perspective on Mandarin Idiom Representation

Le Qiu[1]*, Emmanuele Chersoni[1], Aline Villavicencio[2,3,4]
[1]The Hong Kong Polytechnic University, Hong Kong SAR
[2]The University of Exeter, UK [3]The University of Sheffield, UK
[4]Federal University of Rio Grande do Norte, Brazil

Abstract

Chengyu, or four-character idioms, are ubiquitous in both spoken and written Chinese. Despite their importance, *chengyu* are often under-explored in NLP tasks, and existing evaluation frameworks remain limited in scope and depth. In this paper, we introduce an intrinsic evaluation task for Chinese idiomatic understanding: idiomatic semantic textual similarity (iSTS), which evaluates how well models can capture the semantic similarity of sentences containing idioms. To this purpose, we present a curated dataset: ChengyuSTS. Our experiments show that current pre-trained sentence Transformer models generally fail to capture the idiomaticity of *chengyu* in a zero-shot setting. We then show results of fine-tuned models using the SimCSE contrastive learning framework, which demonstrate promising results for handling idiomatic expressions. We also presented the results of DeepSeek for reference [1]

1 Introduction

Among multi-word expressions (MWEs), idiomatic expressions (IEs) are distinctive as they are often non-compositional, suggesting their meanings may not be straightforwardly derived from individual words (Baldwin and Kim, 2010; Zeng and Bhat, 2021). For example, the phrase *spill the beans* can be interpreted either literally as *tipping over the pulse*, or figuratively as *revealing a secret*. The mix of compositionality and non-compositionality of IEs has imposed significant challenges on various natural language processing (NLP) tasks, including sentiment analysis, machine translation, and more. Proper representation of IEs hence holds significant impacts on linguistics and NLP research.

The present study brings its attention to IEs in the Chinese language or 成语 (*chengyu*), which literally mean *set phrases* and are often referred to as *Chinese idioms* or *four-character idioms*, as they're prototypically exhibited to be collocations of four characters (*chengyu* are not always made up with four characters, nor four-character combinations are necessarily *chengyu*). While Chinese idioms can include various forms of IEs, such as slangs (俚语, *liyu*), sayings (俗语, *suyu*), proverbs (谚语, *yanyu*), and more, the term *Chinese idiom* in this study refers exclusively to *chengyu*.

	画蛇添足
Literal:	to draw a snake and add feet to it
Figurative 1:	to ruin the effects by adding something superfluous
Figurative 2:	to fabricate something out of thin air

Table 1: A prototypical example of *chengyu*: 画蛇添足. It originates from an ancient fable: In a snake-drawing contest, a man finished first but tried to improve it by adding extra legs. He ended up ruining the drawing and lost to another contestant, who kept his snake simple and unadorned. The phrase 画 (*to draw*) 蛇 (*snake*) 添 (*to add*) 足 (*foot*) compositionally outlines the story, and two extra figurative meanings have been derived on the basis.

In contrast to idioms in Western languages, which usually resemble regular phrasal constructions (Cacciari, 2014), Chinese idioms frequently deviate from the syntactic norms of modern Mandarin. They are typically compact and synthetic in structure, and exhibit opacity in semantics (Tsou, 2012; Liu and Su, 2021), as shown in Table 1. This characteristic adds to their expressive richness but poses challenges for interpretation without adequate socio cultural knowledge.

Researchers have made efforts to improve IE representations. For instance, Zeng and Bhat (2023) have created context-aware representations for English PIEs, by unifying BART's (Lewis, 2019) ability to generate compositional meanings with an *idiomatic adapter* that captures their non-compositional meanings (Zeng and Bhat, 2022).

---

[*] Correspondence: lani.qiu@connect.polyu.hk

[1] Data and code are available upon `https://github.com/Laniqiu/ChengyuSTS`.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 1–12
November 8-9, 2025 ©2025 Association for Computational Linguistics

For Chinese IEs, Wu et al. (2024) have proposed to boost idiomatic representations by mitigating the inconsistency between different semantic representations using multi-semantic contrastive learning. However, evaluations of Chinese idiomatic representations so far have been carried out in more limited settings, compared to English studies. Idiomaticity representation can be accessed extrinsically, through performance on downstream tasks such as machine translation, or intrinsically, by probing the linguistic properties encoded within the representations (Zeng and Bhat, 2023; He et al., 2024a). While studies on English IEs leverage both extrinsic and intrinsic methods (e.g. Zeng and Bhat, 2021, 2022, 2023), varying in task genres, work on Chinese idioms has largely depended on cloze-style tasks (e.g. Long et al., 2020; Wu et al., 2024). While such tasks directly assess language ability and are intrinsic in nature, they emphasize specific contextual use and are thus relatively less intrinsic than evaluations based on STS or embedding analysis, which focus more on semantic structure and are therefore reflect a deeper level of intrinsic evaluation. For simplicity, we refer to the former as *relatively extrinsic*, without implying the standard extrinsic evaluation based on downstream tasks.

The observed limitations arise from multiple aspects, including a lack of diverse evaluation methods and datasets. In light of this, the study takes the initiative to explore the intrinsic evaluation of idiomatic representation in Mandarin. Inspired by Tayyar Madabushi et al. (2022), we adopt the iSTS task (see section 3) as an intrinsic method and present an STS (semantic textual similarity) benchmark on Chinese idioms: **ChengyuSTS**. The dataset could provide a novel alternative for assessing Chinese idiom representations from an intrinsic perspective. Additionally, we investigated the performance of several Transformer models and DeepSeek (DeepSeek-AI, 2025) on the dataset. To our knowledge, this is the first evaluation of this new LLM (large language model) on Chinese idiomatic representations. Plus, we introduce several baseline models trained with contrastive learning, with the best ones yielding results comparable to DeepSeek.

The following sections are organized as follows: Section 2 introduces some related work and section 3 details the task setup and dataset creation. Section 4 presents the evaluation setup and results. Finally, section 5 concludes the current study and suggests directions for future work.

2 Related Work

2.1 MWE Representation

MWEs, such as IEs, have been a longstanding hard-nut for NLP (Sag et al., 2002; Liu et al., 2017; Shwartz and Dagan, 2019; Biddle et al., 2020; Zeng and Bhat, 2022). Studies have suggested that PLMs (pre-trained language models) such as BERT (in its basic version) cannot model idiomaticity properly (Shwartz and Dagan, 2019; Wu et al., 2024). Even ChatGPT models (Brown, 2020), as powerful as they can be, have exhibited limitations in interpreting IEs (Raunak et al., 2023).

Early attempts to model MWEs, including IEs, generally employed either a *distributional* or a *compositional* approach. The distributional approach treats the entire phrase as a single, inseparable unit, learning its embedding from the context, much like the process for individual word embeddings (Mikolov, 2013; Yin and Schütze, 2014, 2016). While effective, this approach often requires large-scale corpora and thus is prone to data sparsity issues. On the other hand, the compositional approach constructs phrase embeddings by aggregating the embeddings of constituent words (Mitchell and Lapata, 2010; Yu and Dredze, 2015), but it struggles to represent non-compositional phrases such as IEs. Given these limitations, hybrid approaches have been developed to jointly learn phrase embeddings (Hashimoto and Tsuruoka, 2016; Li et al., 2018a,b).

Recent work has sought PLMs for IE representation using adaptive and contrastive learning techniques. Zeng and Bhat (2022) pioneered this direction by introducing GIEA, a contextualized embedding model that captures non-compositional semantics with an *idiomatic adapter* added to BART (Lewis, 2019). Building on this, they further developed PIER (Zeng and Bhat, 2023), a refined model that has been shown to effectively balance compositional and non-compositional representations of IEs. Additionally, He et al. (2024a) and Wu et al. (2024) integrated contrastive loss into their methods, respectively, enabling the models to better distinguish between different semantic interpretations. Studies have also found that incorporating external linguistic knowledge, such as hypernymy, synonyms or definitions can enhance model performance (Long et al., 2020; Wang et al., 2020; Sha et al., 2023).

2.2 Representation Evaluation

Word embeddings, by design, represent words as vectors, such that the proximity between vectors reflects semantic similarity relationships between the corresponding words (Schnabel et al., 2015; Bakarov, 2018). Their evaluation can be categorized into extrinsic and intrinsic methods.

Extrinsic methods evaluate representations based on their contribution to downstream tasks, such as machine translation, sentiment analysis or natural language inference, using the task-specific metrics as indicators of quality (Chiu et al., 2016; Zhou et al., 2024). However, the final performance can be influenced by several confounding factors, in addition to evaluating idiomatic accuracy. In contrast, intrinsic evaluation directly examines embeddings, often assessing their alignment with human judgments regarding *similarity* or *relatedness* between words (Schnabel et al., 2015; Chiu et al., 2016; Tsvetkov et al., 2016). By approximating these relevant tasks (e.g., to examine semantic similarity), intrinsic methods evaluate the general properties of word embeddings, without the need to perform each task of the sort (Tsvetkov et al., 2016) and indeed they represent a popular choice for analyzing the linguistic knowledge encoded in embeddings (Lenci et al., 2023; A et al., 2024; Ascari et al., 2024). Besides, they could provide insights into the traits that influence model performance in downstream tasks, and thus guide targeted improvements. As an example of intrinsic evaluation targeting compositionality, Senaldi et al. (2016) built a dataset of Italian verbal idioms and compositional expressions, and then built lexical variants of their items by replacing some of their constituents with semantically-related words. Their results showed that idioms have a lower similarity to their lexical variants in distributional embedding spaces, compared to compositional verb phrases.

On IE representations specifically, extant work adopts both extrinsic (e.g. Škvorc et al., 2022; Chakrabarty et al., 2022) and intrinsic methods (e.g. Dankers et al., 2022; He et al., 2024a). However, datasets that could be leveraged for intrinsic evaluations are mostly available in Western languages, including English, Portuguese and Galician, such as MAGPIE (Haagsma et al., 2020), AStitchInLanguageModels (Tayyar Madabushi et al., 2021), NCTTI Garcia et al. (2021a), and data from SemEval2022 (Tayyar Madabushi et al., 2022). For instance, Zeng and Bhat (2022, 2023) discussed their intrinsic evaluation tasks — *embedding clustering* and *embedding differentiation* — using the MAGPIE dataset. The clustering task involves grouping IE embeddings into clusters and examining the homogeneity within each cluster to determine if the model produces high-quality embeddings for PIEs that share similar meanings. The differentiation task, on the other hand, assesses a model's ability to distinguish between the literal and idiomatic meanings of the same PIEs within the given context. In a separate stream, He et al. (2024a), focusing on noun compounds (NC) in MWEs, presented their dataset NCIMP for probing the idiomaticity of NCs in English and Portuguese. Besides, they extend the probes from previous work such as Garcia et al. (2021b) and Klubička et al. (2023) and propose a set of measures: *Affinity* and *Scaled Similarity*. Both are relevant to similarity measuring. Affinity examines if an NC is more similar to its synonyms than to other semantically related targets and distractors, while Scaled Similarity magnifies the similarities in a specific vector space by introducing a rescaling factor (e.g., a random item as the lower bound).

In contrast, Chinese idiom datasets such as ChID (Zheng et al., 2019), PETCI (Tang, 2022), CCT (Jiang et al., 2018), CIP (Qiang et al., 2023), and IDIOMKB (Li et al., 2024), are primarily constructed to examine idiom behaviors in applied tasks, including cloze tests, translation, and paraphrasing. CHENGYU-BENCH (Fu et al., 2025), a most recent benchmark, integrates Chengyu-Bench, encompasses three distinct tasks. However, its core focus remains on evaluating surface-level contextual competence, rather than deeper semantic understanding. These resources often lack fine-grained annotations on idiom usage, semantic similarity, or contextual flexibility, making them less suitable for deeper intrinsic exploration. We argue that this limitation contributes to the scarcity of idiom-focused intrinsic evaluation frameworks in current research.

2.3 Idiomatic Semantic Textual Similarity

STS has long been a prominent area in the linguistics and NLP communities. Notable studies revolve around the general domain, such as OCNLI (Hu et al., 2020), and STS benchmarks (Agirre et al., 2016, etc.), yet few are dedicated to the idiomatic STS (iSTS) in the Mandarin language.

The iSTS task, initially introduced by Tayyar Madabushi et al. (2021), is rooted in such a theory: if a model accurately encodes an MWE, the embedding of a sentence containing the MWE should be semantically similar to the embedding of the same sentence where the MWE is substituted with another phrase. Two cases arise depending on the substitution:

1. The MWE has been paraphrased appropriately. In this case, the sentence pair $P(S, A_c)$, where S is the original sentence, A_c is derived from S by correctly paraphrasing the target MWE, should have a semantic similarity approximating 1.
2. The MWE has been replaced inappropriately. In this case, two sentence pairs $P(S, A_i)$ and $P(A_c, A_i)$, where A_i is derived from incorrect paraphrasing, should have roughly the same similarities (see Table 2 for demonstration).

This iSTS setting can be formulated as in Equation 1. It assesses if a model genuinely captures the meaning of an MWE by evaluating its ability to identify semantically similar/ dissimilar sentence pairs. A major strength is that it doesn't require human-annotated scores for incorrect sentence pairs, making it highly efficient for data annotation. The method was adopted by SemEval 2022 (Tayyar Madabushi et al., 2022; He et al., 2024a,b) as an intrinsic alternative for MWE evaluation on non-Chinese datasets.

$$\forall_{i\in I,\, c\in C} \begin{array}{c} Sim(S, A_c) \approx 1; \\ Sim(S, A_i) \approx Sim(A_c, A_i) \end{array} \qquad (1)$$

The current study has the goal of the intrinsic evaluation of the representation of Chinese idioms, to provide a benchmark that is independent of downstream task performance and to diversify evaluation in idiom research.

3 Chinese Idiomatic STS

3.1 Task Statement

We adapted the iSTS framework (see section 2.3) for the intrinsic evaluation of Chinese idioms and created a new dataset: ChengyuSTS, inspired by the work of Tayyar Madabushi et al. (2021, 2022). ChengyuSTS requires constructing sentence pairs via paraphrasing or replacements. The replacements can take varied forms:

For a correct replacement, a Chinese idiom can be substituted with a synonymous or a near-synonymous unit, which could be a single word, an idiom, or a phrase. Alternatively, it can be paraphrased using an explanatory expression that conveys its meaning or a literal interpretation that allows the figurative meaning of the idiom to be inferred. Conversely, incorrect replacements may involve words, idioms, or phrases that are antonymous, contextually inappropriate, or distort the sentence's original meaning. In this sense, the replacements could also be idiomatic. Examples can be found in Table 3.

The subsequent sections detail the dataset creation pipeline.

3.2 Sample Selection

The idiom vocabulary and raw samples are collected from existing resources, including ChID (Zheng et al., 2019), CIP (Qiang et al., 2023) and CCT (Jiang et al., 2018). ChID holds a large collection of four-character Chinese idioms and serves as a gold benchmark for the task of Chinese idiom cloze reading comprehension. The CIP dataset is converted from a machine translation corpus – WMT18 (Bojar et al., 2018) and the CCT dataset contains idioms and sentences crawled online. Given that the texts in ChID are primarily long paragraphs extracted from formal sources (novels, essays and news articles), the inclusion of CIP and CCT could contribute to the diversity of the ChengyuSTS dataset.

The idiom vocabulary was primarily constructed using ChID, as its idioms have been specially collected and filtered based on frequency. We also ensured that the selected idioms are included in Xinhua Dictionary[2], an authorized Chinese-language dictionary that contains over 45,000 idioms with rich meta information. For each idiom, about 3 contexts were sampled across all corpora. We excluded those sentences where the target idiom appears more than once or is only mentioned or referred to. For example, in 他在演讲中用到了“画蛇添足”这个词。(*He used the word of "hua she tian zu" in his speech.*), the idiom 画蛇添足 is simply referred to, instead of being used in a real context. We try to exclude such instances. Also, to minimize the noise from excessively long texts, the text length is restricted to [20, 70], with ChID samples truncated to fit within this range.

---

[2] `https://github.com/pwxcoo/chinese-xinhua.git`

S	A_c	A_i	Expectation
It's a **blood bath**.	It's a **massacre**.	It's a **sanguine fluid bath**.	$Sim(S, A_c) = 1$; $Sim(S, A_i) = Sim(A_c, A_i)$
I feared that taking it would make me a **guinea pig**.	I feared that taking it would make me a **test subject**.	I feared that taking it would make me a **pig**.	

Table 2: Examples of the iSTS data from Tayyar Madabushi et al. (2022). MWEs and their replacements are in **bold**. For brevity, sentences are shortened.

	Original sentence	Correct replacement
E1	巴士的乘客被撞死，火车中的旅客却{安然无恙}。 *Passengers on the bus were killed in the crash, while the passengers on the train were {safe and sound}.*	巴士的乘客被撞死，火车中的旅客却{毫发无损}。 *Passengers on the bus were killed in the crash, while the passengers on the train were {completely unscathed}.*
E2	这些规定{朝三暮四}，叫人无所适从。 *These regulations are {full of chop and change}, leaving people at loss.*	这些规定{经常变动}，叫人无所适从。 *These regulations are {consistently changing}, leaving people at loss.*
E3	你会让这个{稳如泰山}的星座苦恼不安。 *You will disturb this zodiac that is {unshakable as Mount Thai}.*	你会让这个{像泰山一样稳固}的星座苦恼不安。 *You will disturb this zodiac that is {as solid as Mount Thai}.*

Table 3: Examples of paraphrasing, with target idioms and their replacements enclosed in *{}*. Only samples of correct replacement are presented: E1 illustrates the synonym rewording, E2 provides an explanatory substitution, and E3 presents a literal interpretation. Note that these categories are not strictly distinct, as a synonymous phrase may also be an explanation, etc.

3.3 Paraphrasing

For each raw sentence from section 3.2, we construct its homogeneous and adversarial samples via correct and incorrect replacements, as outlined in section 3.1.

In practice, the construction is a collaborative effort between humans and AI. We aim to ensure that the paraphrasing process is dynamic and context-dependent, instead of mechanically substituting an idiom with a set, pre-determined word/ phrase. The use of AI, besides reducing the annotation workload, brings more diversity to the process, thanks to its stochasticity.

Initially, ChatGPT (OpenAI, 2024) is used to paraphrase the target idiom in each sentence using a given candidate, while retaining grammaticality or coherence (an example prompt can be found in Table 4). These candidate replacements, including definitions, synonyms, and antonyms, are extracted from Xinhua Dictionary. Two human experts – both native Mandarin speakers with a Master's degree in linguistics – then compare and review the AI-generated results. Sentences with inappropriate paraphrasing are either revised or discarded. In case of an incorrect replacement, the generated sentence might be slightly incorrect in grammar, which we deem acceptable considering that this is a result of manipulation.

This context-aware and flexible paraphrasing process can produce sentence instances that integrate both the static meaning of an idiom (its dictionary definition, literal or figurative) and its dynamic interpretation (its meaning as shaped by context). Sometimes, the replacements may be morphological variants of the original idioms, through reorganization, abbreviation, or modification, such as 总而言之 -> 言而总之 (*in brief*); 必恭必敬 -> 恭敬 (*being respectful*); 故步自封-> 固步自封 (*being conservative*). By analyzing these sentence pairs, we can better examine whether a model has truly learned an idiom as a holistic unit and grasped the full range of its meaning, rather than merely memorizing its components or capturing a context-insensitive meaning.

3.4 Final Corpus

A fine-tuned model may *cheat* in evaluation by simply assigning a perfect 1 to any sentence pair (Tayyar Madabushi et al., 2021). To prevent this, we deliberately spiced the final corpus with standard STS data. Note that the paraphrased sentences from section 3.3 may exhibit high lexical and structural similarity to their original, while regular STS data may not. For this reason, we chose not to use common datasets such as the Chinese STS-B (Cer et al., 2017), as they don't meet our criteria. Instead, we retrieved about 6,000 Mandarin sentence pairs from PAWS-X (Yang et al., 2019a) and LCQMC

请对以下句子中的成语部分进行改写，尽量保持句子原意和结构不变，并保持改写后的句子语法正确、自然。 句子：他总是对牛弹琴，没人听得懂他的专业术语。 Idiom：对牛弹琴
Please paraphrase only the idiom in the following Chinese sentence. The paraphrased version should preserve the original meaning and structure as much as possible, and be grammatically correct and natural. Sentence: *He is always cating pearls before swine — no one understands his technical jargon.* Idiom: 对牛弹琴 (*to cast pearls before swine*)

Table 4: An example prompt for paraphrasing using ChatGPT.

(Liu et al., 2018), prioritizing those sharing similar structure and wording. Detailed statistics of the final corpus are provided in Table 5 and Table 6 displays some data examples.

	Idiom	Homo.	Advrl.	Total
Train	3,452	9,129 (1,911)	8,898 (2,223)	18,027 (4,134)
Dev	1,219	1,968 (491)	1,895 (515)	3,863 (1,006)
Test	1,153	1,853 (460)	2,010 (549)	3,863 (1,009)
All	3,452	12,950 (2,862)	12,803 (3,287)	25,753 (6,149)

Table 5: Statistical details of the ChengyuSTS dataset. Numbers enclosed in *()* correspond specifically to the counts from the standard STS data. *Homo.* represents a homogeneous sentence pair, e.g., (S, A_c), and *Avrl.* denotes an adversarial sentence pair, i.e., (S, A_i) or (A_c, A_i).

4 Experiments

To establish baseline performance for the ChengyuSTS dataset, we use PLMs to generate sentence embeddings for each sentence pair and then evaluate their alignment with annotations by measuring their pairwise cosine similarity.

We first evaluated the performance of current models on the ChengyuSTS dataset in a *zero-shot* setting in section 4.1, where a model was tested without continuing pre-training or fine-tuning on the ChengyuSTS data. Subsequently, we presented fine-tuned models under the contrastive learning framework in section 4.2. The performance was assessed with the Spearman correlation coefficient, following Tayyar Madabushi et al. (2022); He et al. (2024a,b). [3]

4.1 Zero-shot Evaluation

Importantly, the sentence representations produced by most PLMs, such as BERT (Devlin et al., 2018), are not inherently suitable for the iSTS task due to *anisotropy*(Gao et al., 2019; Li et al., 2020), meaning that the embeddings may not be distributed uniformly across the entire space but rather concentrate within a hypercone, leading to a lack of semantic isometry of the embedding space and introducing biases in cosine similarity measurements (Gao et al., 2019). As evidenced by Reimers and Gurevych (2019), sentence embeddings generated by simply averaging word embeddings or using the CLS-token output from a Transformer demonstrate poor performance in STS tasks. Given this, we sought PLMs that are specifically tailored for sentence representations and are available in Mandarin[4]:

- **Chinese-SBERT**$_{general}$: a Mandarin-adapted sentence Transformer trained for Mandarin STS task.
- **Chinese-SBERT** (Reimers and Gurevych, 2019; Zhao et al., 2019, 2023): a Mandarin-adapted sentence Transformer, trained for Mandarin NLI (natural language inference) task.
- **XLM-SBERT** a multilingual Sentence Transformer for clustering or semantic search tasks.
- **XLM-SBERT-MPNet**: a multilingual Sentence Transformer for clustering or semantic search tasks.

---

[3] The experimental details, including prompts and parameters, can be found in the GitHub repo.

[4] Model keys on Hugging Face (sorted in chronological order). sbert-base-chinese-nli, sbert-chinese-general-v2, distiluse-base-multilingual-cased-v1, paraphrase-multilingual-mpnet-base-v2, simcse-roberta-large-zh and promcse-bert-large-zh.

	S	A_c	A_i	sim
E1	有的网友{付之一笑}，同时也有15名网友积极响应... *Some netizens {brushed it off with a smile}, while 15 others responded actively...*	有的网友{一笑了之}，同时也有15名网友积极响应... *Some netizens {laughed it off}, while 15 others responded actively...*	有的网友{大笑不止}，同时也有15名网友积极响应... *Some netizens {couldn't stop laughing}, while 15 others responded actively...*	
E2	这个丑闻有可能使原本大有前途的政治生涯{戛然而止}。 *This scandal could {bring an abrupt end to} what was once a highly promising political career.*	这个丑闻有可能使原本大有前途的政治生涯{突然中断}。 *This scandal could {abruptly interrupt} what was once a highly promising political career.*		1.0
E3	然而，大多数白马皮肤呈粉红色，有些则有蓝眼睛。 *However, most white horses have pink skin, and some have blue eyes.*		然而，大多数粉红马有白色皮肤和一些蓝眼睛。 *However, most pink horses have white skin, and some have blue eyes.*	0.0

Table 6: Examples from the ChengyuSTS dataset. Target idioms and their replacement in iSTS data are denoted by *{}*. The original sentence S and its correct paraphrase A_c share the same translation, and the translation for its incorrect paraphrase A_i might be partially omitted. For instances with an explicit sim label (e.g., E2 and E3), we expect the model to produce a score approximating the label. In other cases (e.g., E1), the model is expected to assign similar scores to (S, A_i) and (A_c, A_i).

	S	A_c
E1	指挥官还制订在全球多个地点发动{先发制人}或者报复性袭击的计划。 *The commander has also formulated plans to launch {preemptive} or retaliatory strikes at multiple locations around the world.*	指挥官还制订在全球多个地点发动{先声夺人}或者报复性袭击的计划。 *The commander has also formulated plans to launch {show-of-force} or retaliatory strikes at multiple locations around the world.*
E2	许巍的歌声给一代人留下了{不可磨灭}的回忆。 *Xu Wei's singing has left an {inedible} memory for a generation.*	许巍的歌声给一代人留下了{不会随时间消逝}的回忆。 *Xu Wei's singing has left a generation with a memory that won't fade with time.*

Table 7: Hard examples: Only homogeneous sentence pairs (i.e., pairs with an expected similarity of 1) are included. These examples receive calculated similarity scores lower than 0.6 from both DeepSeek and Chinese-SBERT. It can be observed that the model struggles to identify similarity when the target idiom is replaced with either a near-synonym (e.g., E1) or a literal interpretation (e.g., E2), suggesting the models' weakness in iSTS.

	Overall	iSTS only
Chinese-SBERT	30.13	47.38
Chinese-SBERT$_{general}$	37.90	42.32
XLM-SBERT	10.59	32.21
XLM-SBERT-MPNet	21.84	35.98
SimCSE-RoBERTa	40.44	58.75
PromCSE-BERT	42.37	56.81
DeepSeek (R1)	**72.67**	**89.89**

Table 8: Spearman correlation scores the *zero-shot* evaluation results on the test set (reported in %). *Overall* indicates scores on the entire test dataset (3,863 instances), while *iSTS* shows the scores solely on the iSTS data (2,834 instances).

- **SimCSE-RoBERTa**: a RoBERTa trained using the SimCSE framework (Gao et al., 2021) for sentence embeddings.

- **PromCSE-BERT**: a BERT trained using the PromCSE framework (Jiang et al., 2022) for sentence embeddings.

Besides, we also presented the evaluation results with DeepSeek (DeepSeek-AI, 2025), a more recent and advanced LLM that has outperformed other LLMs, including GPT-4o (OpenAI, 2024) and Claude-3.5 Sonnet (Anthropic, 2024), in multiple Chinese benchmarks. In practice, we prompted the model to assign a similarity score to each sentence pair within [0, 1], where 0 suggests complete dissimilarity and 1 perfect similarity.[5]

The results are provided in Table 8. DeepSeek achieves a strong performance, while others show their weakness in discriminating similar/ dissimilar sentences when Chinese idioms are replaced, even though they have been (continuing) pre-trained on large-scale corpora for NLI or STS tasks. This suggests that such models fall short of modeling the underlying meanings of idioms (see Table 7).

While the performance of the DeepSeek model can be seen as an upper bound, the adoption of specialized approaches may be necessary to improve the understanding of less powerful models like PLMs. In the next section, we propose an additional experiment with contrastive learning.

4.2 Contrastive Learning to Mandarin Idiomatic Representation

Contrastive learning, which brings similar embeddings closer and pushes dissimilar ones apart, has been suggested to enhance the semantic isometry of embedding spaces. As further evidenced in Table 8, the SimCSE framework exhibits notable advantages in capturing semantic similarity. Motivated by these findings, we fine-tuned models directly on Chengyu-STS using the supervised version of SimCSE (Gao et al., 2021). The training objective of SimCSE within a mini-batch N is defined in Equation 2, where h_i denotes the representation of sample x_i, $sim(h_1, h_2)$ the cosine similarity between h_1 and h_2, and τ is a temperature hyperparameter.

$$-\log \frac{e^{sim(h_i,h_i^+)}/\tau}{\sum_{j=1}^{N}(e^{sim(h_i,h_j^+)/\tau}+e^{sim(h_i,h_j^-)/\tau})} \quad (2)$$

The key aspect of this approach is to construct a triplet (x_i, x_i^+, x_i^-) for instance x_i, where x_i^+ is semantically similar to x_i and x_i^- is dissimilar. In our experiments, we used the correctly paraphrased instances of x_i as positive samples x_i^+, and their incorrect paraphrases as negative samples x_i^-. If an x_i^+ is unavailable, we duplicate x_i as its positive sample and in the case of a missing x_i^-, we sample a random instance from the dataset as a substitute.

In zero-shot evaluation, only a limited range of models are available. However, for fine-tuning, we explored a broader variety of base models, including Mandarin-adapted BERT, RoBERTa, and XLNet (Yang et al., 2019b; Cui et al., 2019, 2020)[6]. The results, obtained in 4 training epochs, are reported in Table 9.

	Overall	iSTS only
BERT-Chinese	60.03	82.81
Chinese-BERT-WWM	**63.65**	83.95
Chinese-RoBERTa-WWM	63.28	**84.40**
Chinese-XLNet	44.50	79.87

Table 9: Spearman correlation scores of the *fine-tuning* experiments results on the test set (reported in %). *WWM* denotes whole-word-masking, a pre-training strategy (Cui et al., 2019).

The *fine-tuning* results present a great improvement over the zero-shot evaluation. The contrast between them suggests that current sentence Transformers struggle to model idiomaticity, without specific adaptation. The fact that Chinese-RoBERTa-WWM (Cui et al., 2020, 2019) receives the best fine-tuning performance on iSTS data may stem from its WWM strategy during the pre-training phase. By masking entire words rather than compositional units, the model is encouraged to learn the representations of complete semantic units, which aligns well with the fixed structures of idiomatic expressions. However, the performance drop on the overall test set may arise from the inherent discrepancy between standard STS tasks and iSTS tasks (requiring figurative semantic reasoning), indicating that the PLM's generalization ability is still limited.

While the SimCSE-fine-tuned models still fall short compared to DeepSeek, the improvements they achieved remain meaningful, considering the gaps between them in model scale (e.g., DeepSeek has about 7B parameters while Chinese-RoBERTa-WWM only has about 100M[7]) and data volume. The performance gap could be further bridged in the future with other lightweight strategies.

5 Conclusion and Future Work

Chengyu idioms are an essential component of Chinese linguistics and cultural heritage, yet their computational processing remains underexplored in NLP, due to limited task-specific datasets and evaluation frameworks.

In this paper, we have introduced iSTS (Tayyar Madabushi et al., 2021), the idiom-aware se-

---

[5] We didn't compare other LLMs, as DeepSeek has already demonstrated superior performance on multiple Chinese benchmarks, and that evaluation with LLMs is not the major goal of the study.

[6] Model keys on the Hugging Face Hub (sorted in chronological order): bert-base-chinese, chinese-bert-wwm-ext, chinese-roberta-wwm-ext, and chinese-xlnet-base.

[7] Information obtained by using the *Transformers* library.

mantic textual similarity task, into Chinese, accompanied by a curated dataset ChengyuSTS. Our experiments reveal that pre-trained sentence Transformer models fail to capture idiomaticity in Mandarin under the zero-shot setting, and we presented fine-tuned models using SimCSE (Gao et al., 2021), which significantly improved the performance.

Finally, our work is the first presenting evaluation results for the DeepSeek model on Chinese idiom representation. DeepSeek achieved by far the strongest performance, and it will likely represent the reference for future task approaches.

Limitations

We identify the following limitations in the study:

Data. The paraphrased sentences could be rigid in syntax or inappropriate in grammar, especially in the incorrect replacement scenario. Also, due to time and resource constraints, we could only source limited amounts of standard STS data from existing datasets rather than constructing one. The differences between these datasets and our iSTS objectives may cause performance discrepancies (as seen in Table 8 and 9).

Experiments. The limited variety of Sentence Transformers and CSE models on Mandarin restricts our exploration under the *zero-shot* setting. And our *fine-tuning* experiments only employ the SimCSE framework. Future work should incorporate more diverse model families and contrastive learning paradigms.

Acknowledgements

EC was supported by a GRF grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15612222). AV was partly supported by MRC-FAPESP AIM-Health project.

Ethical Considerations

The datasets and codes used in the study are publicly available, and we strictly followed the terms of use specified by their original providers. The annotators were provided with informed consent and were allowed to withdraw freely. AI tools were used in compliance with API terms, with no sensitive data exposure.

References

Pranav A, Yan Cong, Emmanuele Chersoni, Yu-Yin Hsu, Alessandro Lenci, et al. 2024. Comparing static and contextual distributional semantic models on intrinsic tasks: An evaluation on mandarin chinese datasets. In *Proceedings of LREC-COLING*.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation.

Anthropic. 2024. Claude 3.5 sonnet model card addendum.

Roberto Ascari, Anna Giabelli, Lorenzo Malandri, Fabio Mercorio, and Mario Mezzanzanica. 2024. A fistful of vectors: a tool for intrinsic evaluation of word embeddings. *Cognitive Computation*, 16(3):949–963.

Amir Bakarov. 2018. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. *Handbook of natural language processing*, 2:267–292.

Rhys Biddle, Aditya Joshi, Shaowu Liu, Cecile Paris, and Guandong Xu. 2020. Leveraging sentiment distributions to distinguish figurative from literal health reports on twitter. In *Proceedings of the web conference 2020*, pages 1217–1227.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Cristina Cacciari. 2014. Processing multiword idiomatic strings: Many words in one? *The Mental Lexicon*, 9(2):267–293.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Tuhin Chakrabarty, Yejin Choi, and Vered Shwartz. 2022. It's not rocket science: Interpreting figurative language in narratives. *Transactions of the Association for Computational Linguistics*, 10:589–606.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st workshop on evaluating vector-space representations for NLP*, pages 1–6.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. Can transformer be too compositional? analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yicheng Fu, Zhemin Huang, Liuxin Yang, Yumeng Lu, and Zhongdongming Dai. 2025. Chengyu-bench: Benchmarking large language models for chinese idiom understanding and use.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021a. Assessing the representations of idiomaticity in vector models with a noun compound dataset labeled at type and token levels. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2730–2741, Online. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021b. Probing for idiomaticity in vector space models. In *Proceedings of the 16th conference of the European Chapter of the Association for Computational Linguistics*, pages 3551–3564. Association for Computational Linguistics (ACL).

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. Magpie: A large corpus of potentially idiomatic expressions. In *12th Language Resources and Evaluation Conference: LREC 2020*, pages 279–287. European Language Resources Association (ELRA).

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Adaptive joint learning of compositional and non-compositional phrase embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215, Berlin, Germany. Association for Computational Linguistics.

Wei He, Marco Idiart, Carolina Scarton, and Aline Villavicencio. 2024a. Enhancing idiomatic representation in multiple languages via an adaptive contrastive triplet loss. *arXiv preprint arXiv:2406.15175*.

Wei He, Tiago Kramer Vieira, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2024b. Investigating idiomaticity in word representations. *Computational Linguistics*, pages 1–48.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S Moss. 2020. Ocnli: Original chinese natural language inference. *arXiv preprint arXiv:2010.05444*.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. *arXiv preprint arXiv:2203.06875*.

Zhiying Jiang, Boliang Zhang, Lifu Huang, and Heng Ji. 2018. Chengyu cloze test. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana. Association for Computational Linguistics.

Filip Klubička, Vasudevan Nedumpozhimana, and John D Kelleher. 2023. Idioms, probing and dangerous things: Towards structural probing for idiomaticity in vector space. *arXiv preprint arXiv:2304.14333*.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2023. A comparative evaluation and analysis of three generations of distributional semantic models. *Language resources and evaluation*, 56(4):1269–1313.

M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Bing Li, Xiaochun Yang, Bin Wang, Wei Wang, Wei Cui, and Xianchao Zhang. 2018a. An adaptive hierarchical compositional model for phrase embedding. In *IJCAI*, pages 4144–4151.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Minglei Li, Qin Lu, Dan Xiong, and Yunfei Long. 2018b. Phrase embedding learning based on external and internal context with compositionality constraint. *Knowledge-Based Systems*, 152:107–116.

Shuang Li, Jiangjie Chen, Siyu Yuan, Xinyi Wu, Hao Yang, Shimin Tao, and Yanghua Xiao. 2024. Translate meanings, not just words: Idiomkb's role in optimizing idiomatic translation with language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18554–18563.

Pengfei Liu, Kaiyu Qian, Xipeng Qiu, and Xuan-Jing Huang. 2017. Idiom-aware compositional distributed semantics. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1204–1213.

Te-hsin Liu and Lily I-Wen Su. 2021. Chinese idioms as constructions: Frequency, semantic transparency and their processing. *Language and Linguistics*, 22(4):558–592.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC:a large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Siyu Long, Ran Wang, Kun Tao, Jiali Zeng, and Xinyu Dai. 2020. Synonym knowledge enhanced reader for chinese idiom reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3684–3695.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

OpenAI. 2024. Gpt-4o system card.

Jipeng Qiang, Yang Li, Chaowei Zhang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2023. Chinese idiom paraphrasing. *Transactions of the Association for Computational Linguistics*, 11:740–754.

Vikas Raunak, Arul Menezes, Matt Post, and Hany Hassan. 2023. Do GPTs produce less literal translations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1041–1050, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3*, pages 1–15. Springer.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307.

Marco Silvio Giuseppe Senaldi, Gianluca E Lebani, and Alessandro Lenci. 2016. Lexical variability and compositionality: Investigating idiomaticity with distributional semantic models. In *Proceedings of the ACL Workshop on Multiword Expressions*.

Ying Sha, Mingmin Wu, Zhi Zeng, Xing Ge, Zhongqiang Huang, and Huan Wang. 2023. A prompt-based representation individual enhancement method for chinese idiom reading comprehension. In *International Conference on Database Systems for Advanced Applications*, pages 682–698. Springer.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Tadej Škvorc, Polona Gantar, and Marko Robnik-Šikonja. 2022. Mice: mining idioms with contextual embeddings. *Knowledge-Based Systems*, 235:107606.

Kenan Tang. 2022. Petci: A parallel english translation dataset of chinese idioms. *arXiv e-prints*, pages arXiv–2202.

Harish Tayyar Madabushi, Edward Gow-Smith, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2022. SemEval-2022 task 2: Multilingual idiomaticity detection and sentence embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 107–121, Seattle, United States. Association for Computational Linguistics.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin K Tsou. 2012. Idiomaticity and classical traditions in some east asian languages. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 39–55.

Yulia Tsvetkov, Manaal Faruqui, and Chris Dyer. 2016. Correlation-based intrinsic evaluation of word vector representations. *arXiv preprint arXiv:1606.06710*.

Xinyu Wang, Hongsheng Zhao, Tan Yang, and Hongbo Wang. 2020. Correcting the misuse: A method for the chinese idiom cloze test. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 1–10.

Mingmin Wu, Yuxue Hu, Yongcheng Zhang, Zeng Zhi, Guixin Su, and Ying Sha. 2024. Mitigating idiom inconsistency: A multi-semantic contrastive learning method for chinese idiom reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19243–19251.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wenpeng Yin and Hinrich Schütze. 2014. An exploration of embeddings for generalized phrases. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 41–47.

Wenpeng Yin and Hinrich Schütze. 2016. Discriminative phrase embedding for paraphrase identification. *arXiv preprint arXiv:1604.00503*.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic expression identification using semantic compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.

Ziheng Zeng and Suma Bhat. 2022. Getting bart to ride the idiomatic train: Learning to represent idiomatic expressions. *Transactions of the Association for Computational Linguistics*, 10:1120–1137.

Ziheng Zeng and Suma Bhat. 2023. Unified representation for non-compositional and compositional expressions. *arXiv preprint arXiv:2310.19127*.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

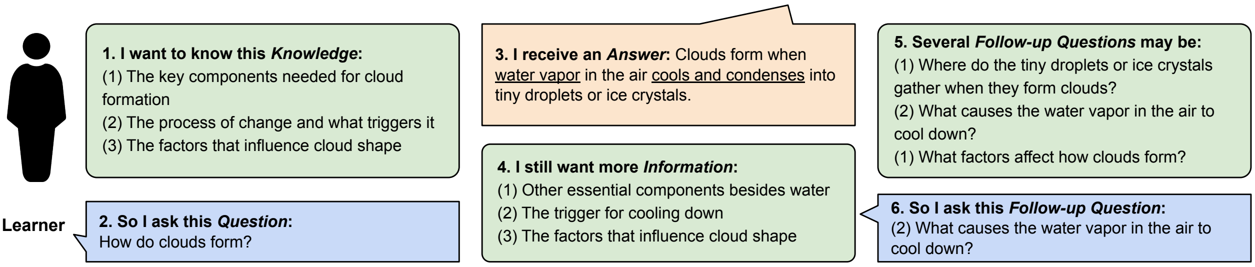
Zhe Zhao, Yudong Li, Cheng Hou, Jing Zhao, et al. 2023. Tencentpretrain: A scalable and flexible toolkit for pre-training models of different modalities. *ACL 2023*, page 217.

Chujie Zheng, Minlie Huang, and Aixin Sun. 2019. Chid: A large-scale chinese idiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 778–787.

He Zhou, Yu Yin Hsu, and Emmanuele Chersoni. 2024. Evaluating Chinese Noun Compound Interpretation in Natural Language Inference. In *Proceedings of the Chinese Lexical Semantics Workshop (CLSW 2024)*.

Bridging Information Gaps with Comprehensive Answers: Improving the Diversity and Informativeness of Follow-Up Questions

Zhe Liu[1*] **Taekyu Kang**[1*] **Haoyu Wang**[1] **Seyed Hossein Alavi**[1,2] **Vered Shwartz**[1,2]
[1] University of British Columbia [2] Vector Insitute
`{zheliu92, salavis, vshwartz}@cs.ubc.ca`
`{davidk15, macdude}@student.ubc.ca`



1. I want to know this *Knowledge*:
(1) The key components needed for cloud formation
(2) The process of change and what triggers it
(3) The factors that influence cloud shape

2. So I ask this *Question*:
How do clouds form?

3. I receive an *Answer*: Clouds form when water vapor in the air cools and condenses into tiny droplets or ice crystals.

4. I still want more *Information*:
(1) Other essential components besides water
(2) The trigger for cooling down
(3) The factors that influence cloud shape

5. Several *Follow-up Questions* may be:
(1) Where do the tiny droplets or ice crystals gather when they form clouds?
(2) What causes the water vapor in the air to cool down?
(1) What factors affect how clouds form?

6. So I ask this *Follow-up Question*:
(2) What causes the water vapor in the air to cool down?

Figure 1: Illustration of a learner's cognitive process in generating follow-up questions.
Green: Implicit cognitive stages; Callouts: Explicit (question, answer, follow-up) triplets collected for the dataset.

Abstract

Generating diverse follow-up questions that uncover missing information remains challenging for conversational agents, particularly when they run on small, locally hosted models. To address this, we develop an *information-gap-driven* knowledge distillation pipeline[1] in which a teacher LLM generates a comprehensive answer, contrasts it with the initial answer to identify information gaps, and formulates gap-bridging follow-up questions. Using this pipeline, we augment the existing FOLLOWUPQG dataset tenfold. We then fine-tune smaller student models on the augmented dataset to distill the teacher's knowledge. Experiments with selected teacher-student model pairs show that fine-tuned students achieve significantly higher informativeness and diversity than variations trained on the original dataset. These findings indicate that our pipeline, which mirrors the human cognitive process of information seeking, provides an efficient distillation channel from state-of-the-art LLMs to smaller models, enabling resource-constrained conversational systems to generate more diverse and informative follow-up questions.

1 Introduction

Asking questions is a fundamental mechanism for humans to acquire new information, particularly when existing information is incomplete. While large language models (LLMs) excel at passively answering questions from users, their ability to proactively guide conversations by identifying and addressing information gaps remains underdeveloped (Liu et al., 2025), with smaller models performing even worse. Therefore, the task of question generation (QG) has become a focal point in natural language processing (NLP) for its role in improving information-seeking dialogue systems (Chen et al., 2024)—including, making information seeking more accurate and efficient (Qi et al., 2020), resolving ambiguities (Li et al., 2017), and ultimately better understanding users' needs to provide suitable assistance across various domains (Laban et al., 2022; Arslan et al., 2024; Li et al., 2024).

While most existing QG tasks focus on generating questions directly answerable from a given context (Zhao et al., 2018; Pan et al., 2020; Ghanem et al., 2022)—a process that diverges from how humans infer and pursue missing information, Meng et al. (2023) propose the FOLLOWUPQG task, which requires models to generate *follow-up questions* that build on, but are not answerable by, the initial question-answer pair. They create the FOLLOWUPQG dataset and show that existing models often produce repetitive or context-bound questions that fail to target unexplored information (Meng et al., 2023). The core challenges of the FOLLOWUPQG task can be formulated into two dimensions: (1) identifying information gaps, the

---

*Denotes Equal Contribution.
[1] Code available at `https://github.com/zheliu92/nlp_followupqg_public/`

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 13–30
November 8-9, 2025 ©2025 Association for Computational Linguistics

unanswered aspects of the initial question, and (2) generating diverse questions that target these gaps.

Building on traditional QG methods (Zhao et al., 2018; Pan et al., 2020; Ghanem et al., 2022), recent work attempts to generate information-seeking follow-up questions using preference optimization (Mazzaccara et al., 2024) and knowledge graphs (Liu et al., 2025), but still lack explicit mechanisms to model gaps or ensure diversity. To address these limitations, we propose an information-gap-driven teacher-student knowledge distillation pipeline. In our approach, a teacher LLM generates a hypothetical "complete" response to the initial question, contrasts it with the often incomplete initial answer to identify information gaps, and formulates gap-bridging follow-up questions. By generating multiple follow-up questions, each targeting some unanswered information, this pipeline ensures the diversity and informativeness of the follow-up questions. For example, in Figure 1, if the initial answer to "*how do clouds form?*" is "*clouds form when water vapor cools,*" a comprehensive answer might add "*… and condenses around dust particles,*" which explicitly exposes the gap through contrast and leads to an informative follow-up question such as "*What role do particles play in cloud formation?*"

Our pipeline can be applied across different teacher-student model pairs. In this work, we use GPT-4o (`2024-02-15-preview`) as the teacher model and BART-large as the student model to verify the pipeline. Specifically, we use GPT-4o to generate the comprehensive answers and follow-up questions. After verifying the quality of the follow-up questions via human evaluation, we then augmented the original FOLLOWUPQG training set tenfold and fine-tuned BART-large on both the original dataset and our augmented dataset. Leveraging GPT-4o to generate high-quality training data, and then distilling the teacher's knowledge into smaller models, our approach achieves strong performance at a significantly lower cost. The experimental results demonstrate significant improvements of the augmented dataset over the baselines, both in terms of quality (validity, relevance, informativeness, etc.) and diversity. Our contributions are as follows:

- We propose an *information-gap-driven* teacher-student knowledge distillation pipeline that generates follow-up questions through contrastive analysis of initial answers and generated comprehensive answers.
- We augment the FOLLOWUPQG training set with over 25,000 high-quality synthetic examples.
- Experimental results show that small models fine-tuned on our augmented dataset outperform peer small-model baselines and achieve near parity with representative LLM-based *Teacher* and *Chain-of-Thought* models.

2 Related Work

Question generation (QG) focuses on automatically generating semantically meaningful and well-structured questions based on a given text (Ali et al., 2010). While traditional QG techniques have made significant strides in domains such as machine comprehension (Du et al., 2017; Uto et al., 2023), e-commerce (Wang et al., 2021), and education (Luo et al., 2024), they primarily generate questions based on known answers. This approach contrasts sharply with human questioning behavior, which actively seeks new information from various perspectives. This limitation has led to the emergence of FOLLOWUPQG, a task whose goal is to generate questions that explore previously unanswered or underexplored aspects of a given text.

FOLLOWUPQG has evolved from simpler methods, such as template-based and retrieval-driven approaches (Kumar and Joshi, 2017; Soni and Roberts, 2019; B et al., 2020), to more advanced techniques that prioritize informativeness (Majumder et al., 2021; Mazzaccara et al., 2024). Knowledge-enhanced approaches, like those in Ge et al. (2023) and Gupta et al. (2022), leverage entity-relation pairs and knowledge graphs to improve the depth of the generated questions. Further advancing this, Liu et al. (2025) combined knowledge graphs with LLMs to increase question informativeness. Efforts to model human-like questioning behavior, such as InquisitiveQG (Ko et al., 2020), have relied on crowd-sourced follow-up questions written for news articles rather than those naturally generated by humans, leading to a lack of depth and cognitive diversity.

We follow the setting of the FOLLOWUPQG (Meng et al., 2023), which formalizes information-seeking follow-up question generation. Based on questions and answers from the ELI5 (explain like I'm 5) subreddit, follow-up questions in this dataset build upon—but are not answerable by—the initial question-answer pair, resembling real-world dialogues where follow-ups resolve ambiguities or deepen understanding.

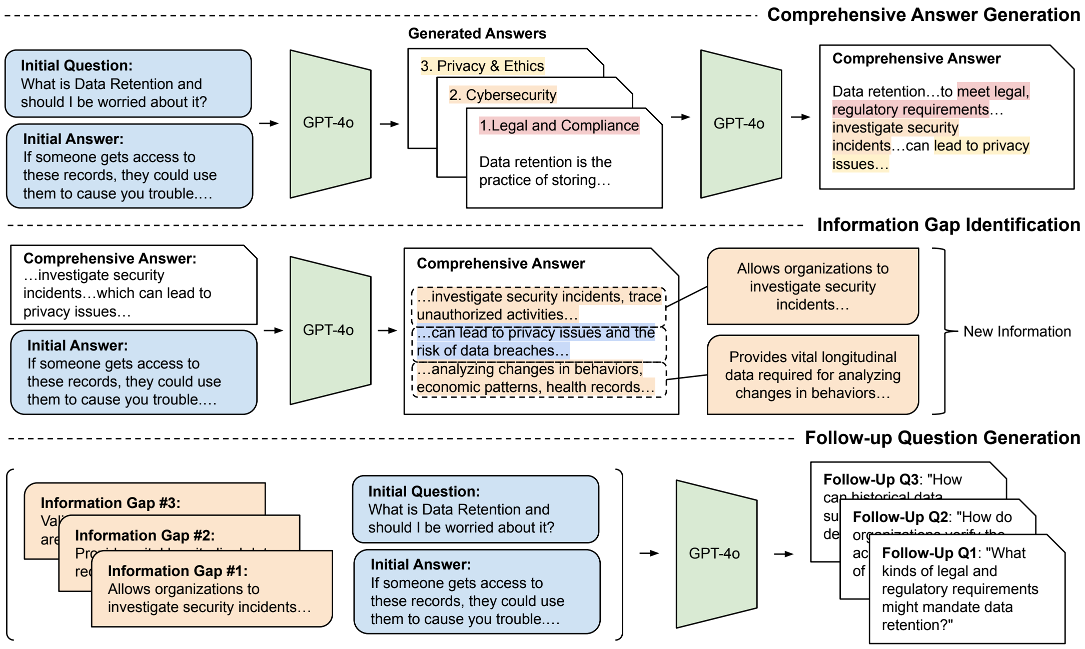

Comprehensive Answer Generation
Initial Question:
What is Data Retention and should I be worried about it?
Initial Answer:
If someone gets access to these records, they could use them to cause you trouble….
GPT-4o
Generated Answers
3. Privacy & Ethics
2. Cybersecurity
1.Legal and Compliance
Data retention is the practice of storing…
GPT-4o
Comprehensive Answer
Data retention…to meet legal, regulatory requirements… investigate security incidents…can lead to privacy issues…
Information Gap Identification
Comprehensive Answer:
…investigate security incidents…which can lead to privacy issues…
Initial Answer:
If someone gets access to these records, they could use them to cause you trouble….
GPT-4o
Comprehensive Answer
…investigate security incidents, trace unauthorized activities…
…can lead to privacy issues and the risk of data breaches…
…analyzing changes in behaviors, economic patterns, health records…
Allows organizations to investigate security incidents…
Provides vital longitudinal data required for analyzing changes in behaviors…
New Information
Follow-up Question Generation
Information Gap #3:
Information Gap #2:
Information Gap #1:
Allows organizations to investigate security incidents…
Initial Question:
What is Data Retention and should I be worried about it?
Initial Answer:
If someone gets access to these records, they could use them to cause you trouble….
GPT-4o
Follow-Up Q3: "How
Follow-Up Q2: "How do
Follow-Up Q1: "What kinds of legal and regulatory requirements might mandate data retention?"

Figure 2: **Data augmentation pipeline.** For a Q&A pair, a comprehensive answer is first generated to the question. By comparing it with the initial answer, information gaps are identified. Finally, multiple follow-up questions are generated targeting those gaps.

Meng et al. (2023) found that models often produce questions that are either repetitive or fail to target unexplored information, thus lacking the cognitive diversity and variability seen in human questioning strategies (Sultan et al., 2020). While follow-up QG has made significant progress, existing approaches largely focus on generating questions directly, using various model architectures and knowledge enhancement techniques (Ge et al., 2023; Liu et al., 2025). Our work, however, takes a novel approach inspired by the human cognitive process that models information gaps and uses them to guide the follow-up question generation.

3 Data Augmentation

Effective FOLLOWUPQG requires models to infer and target gaps between the provided answer and the broader context of a conversation. Following the task definition by Meng et al. (2023): "to generate follow-up questions that seek new information given the initial question and answer", we denote the "initial question" as IQ, "initial answer" as IA, and the "follow-up question" as FQ. We identified critical limitations in the training dataset, including quality issues, which we addressed through dataset cleaning (§3.1). To overcome the small scale (2,790 instances) and low diversity of the dataset, we present a novel data augmentation pipeline (§3.2). Finally, we demonstrate that the augmented dataset retains high quality (§3.3).

3.1 Data Cleaning

The FOLLOWUPQG dataset is limited by its small scale, comprising 3,790 samples: 2,790 for training, 500 for validation, and 500 for testing. Within the 2,790 training instances, there are only 2,651 unique (IQ, IA, FQ) triplets, indicating duplication. Moreover, the dataset consists of 2,648 unique (IQ, IA) pairs, meaning that 99.8% of the (IQ, IA) pairs have only one reference FQ. Training models on this set could thus lead to poor follow-up question diversity. Our further analysis also uncovered data quality issues, likely stemming from automated data collection (see Appendix A). To improve the data quality, we did the following:

- **Deduplication**: We removed 139 duplicate (IQ, IA, FQ) triplets.
- **Reference quality check**: We manually filtered out 84 instances where the reference FQ diverged entirely from the initial question.
- **Sensitive content removal**: We excluded 24 instances involving topics like self-harm or crime, which LLMs are likely to refuse to answer.

The cleaned dataset (2,543 instances) retained broad topic coverage, containing 2,533 unique (IQ, IA) pairs.

3.2 Augmentation Pipeline

As discussed in §3.1, the limited scale of the dataset and the lack of follow-up question diversity hinder the coverage of diverse questioning strategies, restricting model generalization. To address this, we design a GPT-4o-based pipeline that augments the original dataset by generating additional follow-up questions. Our pipeline simulates human reasoning through three interconnected stages: comprehensive answer generation, information gap identification, and follow-up question generation.[2]

Comprehensive answer generation. To identify gaps in the IA, we generate a comprehensive answer (CA) that represents a complete and thorough response to the IQ. As shown in Figure 2, we prompt GPT-4o iteratively to generate answers to IQ that target different perspectives, and synthesize a unified CA. More specifically, GPT-4o was prompted to generate a combination of new and different answers that do not overlap with the other answers, where each answer focuses on a unique aspect not covered in the other generated answers.

Information gap identification. The next step is to identify key concepts or details discussed in the comprehensive answer (CA) but not covered in the initial answer (IA). This is done by prompting GPT-4o. As shown in Figure 2, the initial answer covers the topic of privacy issues but does not cover areas of cyber security (i.e. an information gap). To confirm the validity and reasonableness of the comprehensive answers and identified information gaps, we manually evaluated a random sample of comprehensive answers and ensured that they were accurate and reasonable. Examples of comprehensive answers can be seen in Table 7 and Table 23.

Follow-up question generation. Using the identified information gaps, we prompt GPT-4o to generate follow-up questions that address those gaps while maintaining contextual relevance to the IQ and IA. The generated questions must meet three criteria: be (1) answerable by the CA, (2) unanswerable by the IA, and (3) grounded in terminology and context from the IQ.

After augmentation, each (IQ, IA) pair now includes an average of 10.95 FQs. To preserve the original FOLLOWUPQG format, we automatically remove artifacts such as bullets or numbering from the generated FQs and merge them with the cleaned human-written examples. The resulting dataset comprises 27,874 samples—about 10× the original size—and better reflects the open-ended nature of human questioning, providing models with diverse, explicit signals for addressing information gaps.

3.3 Augmented Data Validation

To assess the quality of the generated follow-up questions, we conducted a human evaluation study on Cloud Connect, using Meng et al. (2023)'s survey. To ensure high-quality annotations, we restricted participation to native English-speaking annotators with a minimum of 1,000 completed annotation tasks and an approval rating exceeding 90%. A randomly sampled subset of 100 (IQ, IA, FQ) triplets was evaluated based on three key criteria: (1) whether the FQ was a valid question,[3] (2) whether any component of the triplet contained sensitive information, and (3) the degree of relatedness between the FQ and the (IQ, IA) pair. The full survey format, including example annotations, is provided in Appendix C. The results show that 94% of the FQs are labeled as valid, 92% as not sensitive, and 91% are related to the original (IQ, IA) pair. Inter-annotator agreement was moderate, with a Cohen's Kappa score of $\kappa = 0.73$ (Cohen, 1960).

4 Experiment Setup

Model Variants. To assess our proposed pipeline and augmented dataset, we fine-tuned BART-large (Lewis et al., 2020) (24 layers, 16 attention heads, hidden size = 1024) on several versions of the FOLLOWUPQG data (Meng et al., 2023), producing three model variants. BART-large is a seq2seq model that conditions on the concatenated IQ and IA to generate an FQ. We chose it as our base model because of its strong performance reported by Meng et al. (2023). As their implementation is not public, we reproduced their training setup (batch = 8, epochs = 10, Adam (Kinga et al., 2015)) and found that the original learning rate of 5e-5 caused instability, so we reduced it to 2e-5; all other hyperparameters remain unchanged.[4]

---

[2] Please refer to Appendix B for the LLM prompts used for the following stages.

[3] Following Meng et al. (2023), a valid question must be in a question format and ask for meaningful information, including Wh-questions (what/why/where/etc.), open-ended questions, probing questions, etc.

[4] Full hyperparameter details and reproduction results are provided in Appendix D.

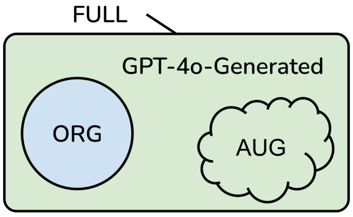


Figure 3: **The ORG/AUG/FULL Dataset.** ORG is the original dataset of ELI5 from the FollowupQG paper. We augment ORG using GPT-4o. FULL combines ORG with all GPT-4o-generated data (about 10× larger). AUG is a random sample of only GPT-4o-generated instances, equal in size to ORG and excluding its data.

We report three variants trained on the FOLLOWUPQG training set as shown in Figure 3: the *ORG* model is trained on the 2,790 original instances from Meng et al. (2023) and serves as our small-model baseline; the *AUG* model, trained on a size-matched random sample of 2,790 GPT-generated questions from our augmentation pipeline (Sec. 3.2) to isolate the effect of data quality; and the *FULL* model, trained on the entire 27,874-instance augmented dataset. All variants share identical hyperparameters and are evaluated on the original FOLLOWUPQG validation and test splits.

Decoding. To generate diverse but contextually relevant follow-up questions, we input the initial question and answer into the model in the following format: `IQ <SEP> IA`, and generate 10 follow-up questions by applying beam search with a beam width of 20, selecting the top 10 candidates. We added a diversity penalty of 10 to encourage unique outputs across the groups and set the temperature to $t = 1.0$ to maintain a balance between diversity and coherence. The maximum length for each generation is set to 1024 tokens. Duplicate generations are removed.

LLM Baselines. Beyond the three BART variants, we evaluated two GPT-4o baselines that represent current large-model performance. The *Teacher* setting executes the full information-gap pipeline from §3.2 on each test instance, returning GPT-4o's comprehensive answer, the identified gaps, and the resulting follow-up questions. The *CoT* setting applies the same chain-of-thought prompt but suppresses all intermediate reasoning, outputting only the follow-up questions. Duplicate generations are removed as in the BART variants.

Model	**Total**	**Ungrammatical**	**Filtered (%)**
ORG	2349	781	33.25
AUG	1895	68	**3.58**
FULL	2061	130	6.31

Table 1: Percentage of filtered-out ungrammatical FQs.

	Diversity			**Length (in token)**			
Model	Distinct-1 (%)	Distinct-2 (%)	Clusters per FQ	Avg.	Shortest	Longest	Std. Dev.
ORG	66.06	91.12	0.651	14.25	3	111	10.13
AUG	**77.36**	94.41	0.857	13.13	4	**24**	**2.98**
FULL	77.09	**94.85**	**0.866**	13.17	4	73	3.77
Teacher	77.00	95.07	0.869	11.96	3	25	3.19
CoT	80.65	96.34	0.878	16.36	5	30	4.02

Table 2: Automatic evaluation of follow-up question generation without human reference.

5 Results

To thoroughly assess the quality of the generated follow-up questions, we employ both automatic evaluation (§5.1) and human evaluation (§5.2). As a first step for both evaluations, we automatically identify and remove ungrammatical questions based on syntactic parsing (see Appendix E for a complete description of the filtering process). Table 1 shows the percentage of ungrammatical questions that were filtered out for each model. *AUG* (3.58%) and *FULL* (6.31%) produce far fewer ungrammatical FQs compared to *ORG* (33.25%), demonstrating their ability to generate more well-formed outputs. We focus the rest of our evaluation on the grammatical questions retained after the filtering.

5.1 Automatic Evaluation

Diversity. We assess the diversity of each set of FQs at the (IQ, IA) level and average the scores across the dataset. First, we report Distinct-n (Li et al., 2016), which measures the average distinct n-gram in the FQs associated with each (IQ, IA) pair. Table 2 shows that *AUG* and *FULL* achieve comparable Distinct-1/2 scores, both exceeding *ORG*. Moreover, the *AUG* Distinct-1/2 scores are comparable to those of the GPT-4o *Teacher* baseline and only slightly lower than those of the advanced *CoT* model.

We also compute a sentence-level diversity score. We embed the FQs using `all-mpnet-base-v2` (Reimers and Gurevych, 2019) and apply agglomerative clustering at a distance threshold of 1.0, normalizing the number of clusters by the num-

Model	BERT	Sent. Sim.	B1	B2	B3	B4	METEOR	ROUGE
ORG	86.28	76.74	40.34	8.49	2.54	1.15	17.57	19.09
AUG	85.72	71.91	32.54	4.02	0.69	0.17	13.84	11.07
FULL	85.74	72.42	32.95	4.19	0.85	0.25	14.16	11.79

Table 3: Automatic evaluation of follow-up question generation with human reference. *ORG* (baseline) performs slightly better.

ber of generated follow-up questions. A score of 1 denotes maximum diversity whereas lower values indicate that questions collapse into the same cluster. Again, Table 2 confirms the trend that our augmentation substantially improves the diversity. Moreover, for both metrics, *AUG* is statistically indistinguishable from the GPT-4o *Teacher* and *CoT* baselines, showing that our pipeline elevates small models to LLM-level diversity on the FollowupQG task.

Average question length. We report the average question length in terms of the number of tokens. We hypothesize that shorter questions are generally more readable. Table 2 lists the average length, shortest and longest follow-ups, and standard deviation (SD). The *ORG* model shows the greatest variation in question length (SD = 10.13). Notably, its longest follow-up (111 tokens) far exceeds *FULL* (73) and *AUG* (24). In contrast, *AUG* is the most consistent (SD = 2.98; max = 24), with *FULL* close behind (SD = 3.77)

Qualitatively examining the generated follow-up questions, we find that *AUG* and *FULL* generally produce concise, well-formed queries, while *ORG* sometimes generates very short, vague prompts (e.g., “So it’s cultural?”). Meanwhile, the longer questions from *ORG* and *FULL* often include extraneous conversational filler. Overall, *AUG* maintains structured, concise outputs for follow-up questions, whereas *FULL* and *ORG* exhibit greater variability, occasionally producing overly long or conversational phrasing. More examples are provided in §6.1 and Appendix G.

Similarity to the references. To compare our results with those obtained by Meng et al. (2023), we perform identical automatic evaluations. We measure lexical overlap with BLEU-1-4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004), and semantic similarity with BERTScore (Zhang* et al., 2020) and an embedding-based cosine score computed with `all-mpnet-base-v2` (Reimers and Gurevych, 2019), following Meng et al. (2023). For each metric, we compute the highest score across all generated follow-ups with the human reference and report the average for the entire dataset. Table 3 shows a consistent advantage for *ORG*. This is expected, as both its training data and the test set come from the original FollowupQG distribution. The lower BLEU scores for *AUG* and *FULL* reflect a common issue in open-ended QG: lexically diverse yet valid questions are under-rewarded by n-gram metrics (Pan et al., 2021). In contrast, the gap between *FULL* and *AUG* is much smaller on BERTScore and embedding similarity, which focus on semantic alignment and discount stylistic differences. Consequently, we turn to human evaluation to capture diversity and nuanced informativeness that automatic metrics may overlook.

5.2 Human Evaluation

We conducted a human evaluation to assess the quality of generated FQs on four criteria: validity, complexity (the level of reasoning required), relevance, and informativeness (Details listed in Table 4). We randomly sampled 30 (IQ, IA) pairs from the FollowupQG testing set and evaluated the generated FQs from all five models.

Human evaluation was conducted on Cloud Connect. To ensure high-quality annotations, we restricted participation to native English-speaking annotators with a minimum of 1,000 completed annotation tasks and an approval rating exceeding 90%. Annotators interacted with a structured evaluation interface (see Appendix H). Each task presented an initial question, its corresponding answer, and a generated FQ candidate. Annotators first assessed whether the FQ was valid. If deemed invalid, they proceeded directly to the next task. Oth-

	Question	Numeric Scale
Validity	Is the FQ question a valid question?	yes (1) / no (0)
	Does the FQ contain any of the following errors?	contains errors: • redundant • repetitive • wrong semantic collocation (1) / no errors (0)
Complexity	Does generating the FQ require reasoning?	complex (3) / moderate (2) / minimal (1) / no (0)
Relevance	How relevant is FQ to the initial question and answer?	strongly (3) / relevant (2) / slightly (1) / not (0)
Informativeness	Does the FQ elicit new information?	a lot (3) / some (2) / little (1) / no (0)

Table 4: The aspects evaluated in the human evaluation with respect to the follow-up question (FQ).

	ORG		*AUG*		*FULL*	
	Mean	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**
Validity	0.7324	0.1964	**0.9065***	0.0849	0.8743	0.1102
Complexity	0.9274	1.0129	**1.4798***	0.9441	1.4454	0.7025
Relevance	1.6236	1.4716	**2.0935***	1.0225	1.7377	1.0269
Informativeness	0.7755	0.9563	**1.4517***	1.1297	1.2951	0.8223

Table 5: Human evaluation scores for each aspect comparing *ORG*, *AUG* and *FULL*. Best results are in bold; only statistically significant results are marked with an asterisk.

erwise, they answered four additional evaluation questions, as detailed in Table 4 (See Appendix F for the complete annotation guidelines). Each task was annotated by 3 annotators, yielding substantial inter-annotator agreement, with an average Cohen's Kappa of $\kappa = 0.77$ (Cohen, 1960).

Table 5 reports the mean and variance of each evaluation criterion for *ORG*, *AUG*, and *FULL*. Overall, *AUG* achieved the best results across all criteria, with a statistically significant difference from the other models (tested with a one-way ANOVA). Over 90% of the FQs generated by *AUG* were considered valid, and these questions were rated as relevant, somewhat informative, and minimally to moderately complex. *FULL* closely follows across aspects, while *ORG* lags behind. The only aspect on which *ORG* closely follows *FULL* is relevance, aligning with the findings of Meng et al. (2023) that current models perform well in maintaining relevance. In sum, the results clearly prefer the questions generated by *AUG*, which excel in validity, complexity, relevance, and informativeness—qualities essential for meaningful follow-up questions.

To gauge how effectively our information-gap pipeline distills LLM knowledge into smaller models, we repeated human evaluation on the GPT-4o *Teacher* and *CoT* baselines described in §4. Table 6 shows that both LLM baselines post slightly higher mean scores than the distilled *AUG* model, yet the difference is statistically significant only for the Relevance metric. On validity, complexity, and informativeness, *AUG*—a BART-large model trained on a random, size-matched subset of our augmented data—remains statistically indistinguishable from the much larger *Teacher* and *CoT* models. These results underscore the strength of our pipeline: by contrasting initial answers with LLM-generated comprehensive answers, it simulates the human information-seeking process and produces synthetic follow-up questions rich in diversity and informativeness. Fine-tuning on this augmented data enables small, locally deployable models to reach teacher-level quality at a fraction of the inference cost, thereby making high-performance FollowupQG feasible on resource-constrained hardware.

	Teacher		*AUG*		*CoT*	
	Mean	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**
Validity	0.9457	0.0514	0.9065	0.0849	**0.9647**	0.0342
Complexity	**1.6227**	0.6604	1.4798	0.9441	1.5725	0.7575
Relevance	2.1240	0.7928	2.0935	1.0225	**2.4863***	0.5500
Informativeness	1.6744	0.6502	1.4517	1.1297	**1.6784**	0.8095

Table 6: Human evaluation scores for each aspect comparing (i) *AUG* with *Teacher* and (ii) *AUG* with *COT* GPT-4o. Best results are in bold; only statistically significant results are marked with an asterisk.

The comparative results across models reveal key insights into the role of data quality versus quantity in the task of FollowupQG. Notably, *AUG*, trained on the same number of instances as *ORG* but consisting solely of GPT-4o-generated, high-quality, reasoning-heavy questions, consistently outperforms both *ORG* and *FULL* across most metrics, yielding greater validity, complexity, relevance, and informativeness. This indicates that data quality is more critical than dataset size. Despite consisting of ten times more training data, *FULL* failed to surpass *AUG*, likely because remaining lower-quality or low-reasoning examples from the original dataset dilute learning. These findings challenge the assumption that larger datasets automatically improve performance and underscore the value of targeted augmentation with strict quality control. Future work should explore strategies to scale data while maintaining rigorous curation to further advance follow-up question generation.

6 Analysis

To further understand the strengths and limitations of our method, we present a qualitative comparison of follow-up questions generated by all models for the same (IQ, IA) pair (§6.1), as well as an analysis of the expected information gain from the generated follow-up questions (§6.2).

6.1 Qualitative Analysis

In Table 7, we compare follow-up questions generated by the *ORG*, *AUG*, and *FULL* models for a given (IQ, IA) pair. Questions from *ORG* are often redundant—e.g., "What is a heuristic?"—or

IQ	How would you explain the concept of a heuristic?
IA	It took me a long time to understand heuristics and *I still don't know if I can ELI5 after a handful of bachelor and graduate degrees.* A heuristic is like a clue or a key to help you solve a problem or remember something. That's the best I got.
CA	A heuristic is a practical approach or mental strategy used for problem-solving and decision-making, particularly when dealing with complex or incomplete information. It serves as a cognitive shortcut, allowing individuals to reach quick and satisfactory solutions without the need for exhaustive analysis or perfect precision. By focusing on the most relevant aspects of a problem and often relying on past experiences, common sense, or general rules of thumb, heuristics simplify the decision-making process and reduce cognitive load. While heuristic methods are not guaranteed to provide the best solution, they are typically sufficient for achieving an immediate goal and are useful in many everyday situations where speed and practicality are prioritized over exhaustive accuracy.
FQ by *ORG*	– *How do you know if you can ELI5?* – *What is a heuristic?* – Thanks for the answer. So, what I am trying to understand is, *what is ELI5?* – *That's the best I got. I'm not a psychologist, but my brain picks out heuristics when I'm stressed and it usually works. I don't know if I can ELI5 after a handful of bachelor and graduate degrees. But I'm still working on it.*
FQ by *AUG*	– What are some examples of cognitive strategies that rely on heuristics? – How does the concept of a heuristic relate to the process of problem-solving? – What role does the hippocampus play in heuristic formation?
FQ by *FULL*	– What are some examples of heuristics used in decision-making? – What role do heuristics play in problem-solving? – How do heuristics *help in segmenting and segmenting information* to specific tasks or objectives? – *I'm not sure I can ELI5 after a handful of bachelor and graduate degrees. I'm not even sure if I can understand ELI4. I know that it's a heuristic. But I don't know if I understand ELII5. Like, I know what a heuristics are. And I know how to use a heymn to solve problems. So I'm asking if you can ELII4?*

Table 7: Example of follow-up question generated by three model variants, with comprehensive answers (ID 3182).

tangential, such as "How do you know if you can ELI5?" to the original responder that mentioned they didn't know if they could explain it to a 5-year-old (ELI5), thus drifting away from the target concept of heuristics. While the *FULL* model yields a wider range of relevant questions and excels in diversity, it occasionally produces tangential or wordy phrasing, for instance, "How do heuristics help in segmenting and segmenting information for specific tasks?", which hurts clarity. By contrast, *AUG* strikes the best balance of informativeness and diversity, offering focused, insightful questions like "What are some examples of cognitive strategies that rely on heuristics?" and "How does the concept of a heuristic relate to the process of problem-solving?". Additional examples can be found in Appendix G.

6.2 Quantifying Information Gain

In §5.2 we asked annotators to rate the informativeness of each follow-up question. We now introduce an automated alternative that requires no human raters, leveraging the GPT-4o "comprehensive answers" (CA; see definition in §3.2). We treat each CA as a proxy for the full body of information relevant to its (IQ, IA) context. An FQ is informative if it (i) cannot be answered from the IA alone—otherwise it adds no new information—and (ii) can be answered from the CA—otherwise it is likely irrelevant. Guided by this rule, we prompt GPT-4o to judge the answerability of every model-generated FQ against both the IA and the corresponding CA.

Model	**Human-INF**	**GPT-INF-All (%)**	**GPT-INF-Sel (%)**
ORG	0.7755	25.17	23.29
AUG	**1.4517**	**36.19**	**35.91**
FULL	1.2951	34.90	32.20

Table 8: Comparison of human-annotated informativeness scores and GPT-evaluated informative percentage across models.

Table 8 corroborates the human evaluation of informativeness: *AUG* produces the largest share of informative questions (36 %), followed by *FULL* (35 %) and *ORG* (25 %). Comparing the GPT-4o labels with human-annotated informativeness scores (§5.1) further validates the automatic method: annotators assigned higher mean scores to FQs the model classified as informative (1.29) than to those it did not (1.07). A two-sample t-test ($p = 0.0011$) confirms the statistical significance, although the effect size is small (Cohen's $d = 0.215$) (Cohen, 2013).

7 Conclusion

In this work, we proposed a novel approach to enhance the diversity and informativeness of follow-up questions by explicitly modeling information gaps via an LLM-generated comprehensive answer. We augmented the original FOLLOWUPQG dataset with GPT-4o and distilled this data into a small, locally deployable BART-large model. Experiments show that our pipeline enables the small model to outperform peer small-model baselines and to perform comparably to GPT-4o baseline models in terms of validity, complexity, relevance, and informativeness—all at a fraction of the inference cost. These results suggest that targeted, high-quality augmentation can be more impactful than merely increasing dataset size. They also demonstrate that our method offers a practical approach for improving information-seeking dialogues—by reducing ambiguities and enhancing LLM responses—even on systems with limited computational resources.

Future work could explore ways to increase follow-up-question diversity while reducing redundancy, and to extend the pipeline to downstream tasks involving multi-turn dialogue. We also encourage research on stronger automated metrics for evaluating question quality, given the high cost of human annotation and the limitations of current automatic measures.

Limitations

We acknowledge several limitations in our work. First, while our CA-based pipeline is effective in knowledge-driven contexts, its applicability to non-knowledge-based conversations, such as opinion-based questions (e.g., “What would you do in such a scenario?”), remains unclear, as the subjective judgment required in these conversations can be difficult for a generated CA to capture. Additionally, although our pipeline prioritizes informativeness, follow-up questions do not always need to introduce new information (Kurkul and Corriveau, 2018)—for example, requests for simpler explanations (e.g., “Can you explain this in an easier-to-understand way?”).

Our work also calls for several future works and expansions. For example, our pipeline can be tested and evaluated on languages besides English, including low-resource languages. Moreover, given the pragmatic applicability of this pipeline and its focus on resource-efficiency, it would be pertinent to evaluate the compute-cost tradeoffs to help users make informed decisions. Lastly, our pipeline’s performance can be evaluated on different combinations of Teacher-Student models as well. In the future, we hope to extend this method to support various types of follow-up questions and integrate it into downstream dialogue-based applications.

Ethical Considerations

All annotators involved in the human evaluation for this research were fairly compensated, with payment rates exceeding the local minimum wage to ensure equitable remuneration for their time and effort. Prior to recruiting annotators, ethical approval was obtained from the research ethics board at the authors’ institution, ensuring that the human evaluation process adhered to ethical guidelines and that no harm was caused to any individual involved. Additionally, the FOLLOWUPQG dataset used in this work is publicly available, and we also released the new data created in this work, including the augmented data and generated comprehensive answers, to promote transparency and reproducibility in future work.

Acknowledgments

This work was funded, in part, by the Vector Institute, Canada CIFAR AI Chairs program, Accelerate Foundation Models Research Program Award from Microsoft, and an NSERC discovery grant.

References

Husam Ali, Yllias Chali, and Sadid A. Hasan. 2010. Automatic question generation from sentences. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 213–218, Montréal, Canada. ATALA.

Banu Arslan, Gokhan Eyupoglu, Semih Korkut, Kenan Ahmet Turkdogan, and Ertugrul Altinbilek. 2024. The accuracy of ai-assisted chatbots on the annual assessment test for emergency medicine residents. *Journal of Medicine, Surgery, and Public Health*, 3:100070.

Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. Automatic follow-up question generation for asynchronous interviews. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 10–20, Santiago de Compostela, Spain. Association for Computational Linguistics.

Yunmo Chen, Tongfei Chen, Harsh Jhamtani, Patrick Xia, Richard Shin, Jason Eisner, and Benjamin Van Durme. 2024. Learning to retrieve iteratively for in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7156–7168, Miami, Florida, USA. Association for Computational Linguistics.

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

Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. routledge.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2023. What should I ask: A knowledge-driven approach for follow-up questions generation in conversational surveys. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 113–124, Hong Kong, China. Association for Computational Linguistics.

Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. Question generation for reading comprehension assessment by modeling how and what to ask. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2131–2146, Dublin, Ireland. Association for Computational Linguistics.

Shrey Gupta, Anmol Agarwal, Manas Gaur, Kaushik Roy, Vignesh Narayanan, Ponnurangam Kumaraguru, and Amit Sheth. 2022. Learning to automate follow-up question generation using process knowledge for depression triage on Reddit posts. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*, pages 137–147, Seattle, USA. Association for Computational Linguistics.

D Kinga, Jimmy Ba Adam, et al. 2015. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5, page 6. San Diego, California;.

Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6544–6555, Online. Association for Computational Linguistics.

Vineet Kumar and Sachindra Joshi. 2017. Incomplete follow-up question resolution using retrieval-based sequence to sequence learning. In *Proceedings of the 40th International ACM Sigir Conference on Research and Development in Information Retrieval*, pages 705–714.

Katelyn E Kurkul and Kathleen H Corriveau. 2018. Question, explanation, follow-up: A mechanism for learning from others? *Child Development*, 89(1):280–294.

Philippe Laban, Chien-Sheng Wu, Lidiya Murakhovs'ka, Wenhao Liu, and Caiming Xiong. 2022. Quiz design task: Helping teachers create quizzes with automated question generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 102–111, Seattle, United States. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *5th International Conference on Learning Representations, ICLR 2017*.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jianyu Liu, Yi Huang, Sheng Bi, Junlan Feng, and Guilin Qi. 2025. From superficial to deep: Integrating external knowledge for follow-up question generation using knowledge graph and LLM. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 828–840, Abu Dhabi, UAE. Association for Computational Linguistics.

Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. 2024. Chain-of-exemplar: Enhancing distractor generation for multimodal educational question generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7978–7993, Bangkok, Thailand. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, Online. Association for Computational Linguistics.

Davide Mazzaccara, Alberto Testoni, and Raffaella Bernardi. 2024. Learning to ask informative questions: Enhancing LLMs with preference optimization and expected information gain. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5064–5074, Miami, Florida, USA. Association for Computational Linguistics.

Yan Meng, Liangming Pan, Yixin Cao, and Min-Yen Kan. 2023. FollowupQG: Towards information-seeking follow-up question generation. In *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)*, pages 252–271, Nusa Dua, Bali. Association for Computational Linguistics.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, and Christopher D. Manning. 2020. Stay hungry, stay focused: Generating informative and specific questions in information-seeking conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 25–40, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sarvesh Soni and Kirk Roberts. 2019. A paraphrase generation system for EHR question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 20–29, Florence, Italy. Association for Computational Linguistics.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. 2020. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. Association for Computational Linguistics.

Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. 2023. Difficulty-controllable neural question generation for reading comprehension using item response theory. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 119–129, Toronto, Canada. Association for Computational Linguistics.

Yongzhen Wang, Kaisong Song, Lidong Bing, and Xiaozhong Liu. 2021. Harvest shopping advice: Neural question generation from multiple information sources in e-commerce. *Neurocomputing*, 433:252–262.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

A Problematic and Unrelated Instance Examples

Tables 9 and 10 show examples of a problematic instance and an unrelated instance from FOLLOWUPQG training set, respectively.

Initial Question	How exactly do you die if you touch electric wires? Do you burn? Suffocate? Heart stop?
Initial Answer	If the current going thru you is around 1 amp, the shock may give you a heart attack. So it's like the hospital shock paddles, only it kills you. If the current is like 10 amps, it probably won't give you a heart attack, and you may survive if your organs aren't fully cooked. Which they definitely could be if the current isn't immediately stopped. And cooked kidneys don't function that good. So to summarize, either heart attack or cooked organs/burned, depending on the current. Edit: To clarify, it takes less than 1 amp to stop your heart, and the current also has to go thru your heart. I just used 1 amp and 10 amps as sample values.

Table 9: Problematic Instance 1696: Self-Harm-Related Content.

Initial Question	How its possible for the light to travel constantly with out stop slowing down from the most old galaxies?
Initial Answer	Okay, an NFT is a NON-FUNGIBLE TOKEN. Let's take the Mona Lisa for example. [...] So what you've done is bought the off-brand version (online) instead of the real thing. Also, the guy you paid doesn't really own the Mona Lisa. The TLDR is that you're buying an online version of a product and it says you own it, but you don't own the real thing.
Follow-up Question	Thank you for this! My question is, what's the appeal? Is it just rich-guy-bragging-rights? [...] With, say, a baseball card you can display it and know that it's a rare item, and someday resell it at (you hope) a profit. Are some of these people just speculating and hoping the 'value' goes up?

Table 10: Unrelated Instance 641.

B LLM Prompts

	Prompts
Initial Answer	"Generate an answer focused on a single perspective only, without any conversational fillers. Do not repeat the question in the answer."
Next Answer	"Please provide a new answer focused on a different perspective, ensuring no overlap with previous answers. Focus on unique aspects or insights not covered earlier, and provide the answer only without any conversational fillers. Do not repeat the question in the answer."
Comprehensive Answer	"Synthesize the following answers into a single, comprehensive response. Integrate the key points and insights from each answer, ensuring a cohesive and well-rounded explanation. The final answer should be thorough and address multiple aspects of the question without unnecessary repetition."

Table 11: Comprehensive Answer Generation Prompts: GPT-4o first generates an answer from a single perspective, then iteratively provides non-overlapping answers from different perspectives, which are finally synthesized into a unified response.

Information Gap Identification & Follow-up Question Generation
"Generate all possible follow-up questions as candidates. These follow-up questions must be related to the original question, but must not be rephrases of the original question. These follow-up questions should be answerable by the complete answer. These follow-up questions should not be answered, covered, or detailed by the original answer, but must target terminologies mentioned in the original answer. Separate each follow-up question with '<sep>'."

Table 12: Follow-up Question Generation Prompt.

C Augmented Data - Human Annotation Guideline

Table 13 presents the job description and annotation questions for our human annotation task.

Job Description
Welcome, and thank you for participating in this text evaluation task! In this job, you'll be helping us verify the quality of follow-up questions generated by GPT. For each task, we will provide you with a pair consisting of a question and answer collected from Reddit's "Explain Like I'm Five" (ELI5) forum. You will be asked to evaluate the quality of the follow-up question generated by GPT. These questions and answers aim to provide layperson-friendly explanations for real-life queries. Here is an example of one task sample: Each task may contain noise, such as invalid follow-up questions, sensitive information, or questions unrelated to the original question or answer. Your role is to help us identify these noisy samples. For each task, you will be shown one triple (question, answer, follow-up question). Carefully review each component and answer the following questions based on your judgment:
Q1: Do you think the follow-up question is a valid question? **A.** Yes **B.** No
Q2: Does the initial question, answer, or follow-up question contain sensitive information? **A.** Yes **B.** No
Q3: Do you think the follow-up question is related to the original question and the answer? **A.** Strongly Related **B.** Related **C.** Slightly Related **D.** Not Related

Table 13: Task description and evaluation questions used for human annotation of augmented data.

C.1 Valid/Invalid Question Guideline

The follow-up question might contain multiple sentences but it should consist of at least one valid question. A valid question must be in a question format and ask meaningful information, including Wh-questions (what/why/where/etc.), open-ended questions, probing questions and etc. Invalid questions like "10000 meters? really?", are often used in conversational speech to express feelings instead of asking for new information. Table 14 contains examples of valid and invalid follow-up questions.

Initial Question: Why is the sea calm in the mornings?	
Initial Answer: There are two types of waves which can turn a flat sea into a rougher one - swell waves and wind waves. Swell waves can arrive at any time of day, but because wind waves are generated by the wind, they only develop when the wind begins to blow steadily. Since wind speeds are often low at night, and increase during the daytime, wind waves often die out during the night, leading to a relatively flat sea (perhaps with swell waves) in the early morning. During the day, the wind waves increase in size as the wind speed increases, leading to a rougher, more choppy, sea surface during the afternoon and evening.	
Valid Follow-up	**Invalid Follow-up**
Why are winds always weak in the morning and very strong during the day?	Isn't it common sense that the sea is calmer in the morning?
Reason	**Reason**
The follow-up question is a "Why" question, asking specific reasons about the change of the winds. Therefore, it is a valid question.	This is a rhetorical question because it does not genuinely seek new information. It implies that the answer is obvious and does not contribute to the discussion.

Table 14: Examples of valid and invalid follow-up questions. For the given initial question and answer, the left column presents a valid follow-up question, while the right column features an invalid one, each accompanied by corresponding reasons below.

C.2 Inappropriate Question Guideline

Examples of racist comments include: "It's credit to your race," "Black people will not understand." Examples of hate speech include: "He should go back to where he comes from," "All Mexicans are rapists." Examples of offensive or rude comments include: "Women are not suitable for working in the IT field," "Gay will never understand." Table 15 contains an example of an inappropriate follow-up question.

Initial Question: Why do people develop eating disorders?	
Initial Answer: Eating disorders are complex mental health conditions influenced by a combination of genetic, psychological, environmental, and social factors. While societal beauty standards and pressures can contribute, eating disorders are not simply about wanting to be thin. Conditions like anorexia, bulimia, and binge-eating disorder involve intricate relationships between self-image, emotional regulation, and biological predispositions. Many individuals with eating disorders struggle with anxiety, depression, or trauma, which can further complicate their relationship with food.	
Inappropriate Follow-up	**Reason**
Why don't people with eating disorders just stop starving themselves and eat normally like everyone else?	This question is dismissive. The phrasing is insensitive and could be harmful to individuals struggling with these conditions.

Table 15: Example of an inappropriate follow-up question for the given initial question and answer, accompanied by corresponding reasons below.

C.3 Relevance Question Guideline

- **Strongly Related**: The follow-up question asks for specific definitions, particular reasons, or meanings directly from the original question and answer.
- **Related**: The follow-up question primarily seeks information from the original question or answer but also brings in additional, new information.
- **Slightly Related**: The follow-up question mainly addresses other cases but has some relevance to the original question or answer.
- **Not Related**: The follow-up question does not relate to the original question or answer.

Table 16 contains follow-up questions with various levels of relevance.

Initial Question: Why do airplanes leave white trails in the sky?	
Initial Answer: Those white trails are called contrails, short for condensation trails. They form when hot exhaust from the airplane's engines mixes with the cold air in the upper atmosphere. The water vapor in the exhaust condenses and freezes into tiny ice crystals, creating the white streaks you see in the sky. The persistence of these trails depends on humidity levels; if the air is dry, the contrail dissipates quickly, but if the air is humid, the contrail can linger for a long time.	
Strongly Related Question Example	**Related Follow-up Question Example**
Why do some contrails last longer than others?	Do contrails have any impact on the environment?
Reason	**Reason**
The follow-up question directly builds on the information provided in the answer, specifically regarding the persistence of contrails. Since the answer already mentions humidity as a factor, this question seeks further clarification, making it strongly related.	This follow-up question extends the topic of contrails by asking about their environmental impact. While the original answer does not discuss environmental effects, the question is still relevant because it builds on the phenomenon explained. Thus, it is considered related.
Slightly Related Question Example	**Not Related Follow-up Question Example**
Why do some airplanes make more noise than others?	What causes volcanoes to erupt?
Reason	**Reason**
The follow-up question is about airplanes, which is the general topic of the original question, but it shifts the focus from contrails to noise. While both topics are related to aviation, the connection between them is weak, making the question only slightly related.	The follow-up question introduces a completely unrelated topic (volcanoes) that has no connection to airplanes, contrails, or atmospheric conditions. Since it does not build on the original question or answer in any way, it is considered not related.

Table 16: Examples of follow-up questions' relevance for the given initial question and answer, accompanied by corresponding reasons below.

D Baseline Reproduce

To establish a baseline, we attempted to reproduce the results of Meng et al. (2023) using the reported parameters, as the original implementation was unavailable. We use BART-large, consisting of 24 layers, 16 attention heads, and a hidden dimension of 1024. The initial learning rate (5e-5) led to training instability, which we mitigated by adjusting it to 2e-5 while keeping other hyperparameters unchanged (batch size: 8, epochs: 10, optimizer: Adam (Kinga et al., 2015)). The training was conducted on an NVIDIA Tesla V100 GPU with 32GB of memory, taking approximately 6 hours per run. We set the random seed to 42. After multiple runs, our reproduced model yielded similar overall performance but with some variation-certain metrics improved while others slightly declined (see Table 17). This model served as the baseline for subsequent experiments.

E Filtering Ungrammatical Follow-Up Questions: Examples and Pseudocode

```
def is_valid_question(FQ, IQ, IA):
    return (
        # confirm that FQ ends with '?'
        contains_question_mark(FQ) and
        # use dependency parsing to
            check for
        # WH-Questions, Yes/No Questions
        # and Rhetorical Questions,
        is_question_dependency_parsing(
            FQ) and
        # checks for meaningless/invalid
             words
        # e.g. <QUS>, <EQT>
        not contains_invalid_word(FQ)
            and
        # checks for n-gram substring
            that is
        # duplicated in both FQ and (IQ/
            IA)
        not contains_duplicate_words(IQ,
             IA, FQ)
    )
```

	B1	B2	B3	B4	METEOR	ROUGE	BERT	Sentence Similarity
Reported ORG	17.22	7.11	3.89	2.61	8.00	13.35	-	-
Reproduced ORG	21.88	3.27	1.62	1.13	14.76	12.39	84.41	70.73

Table 17: Comparison of Results from Paper Report and Our Reproduction.

Error Type	**Example**
Improper delimiter	*> Constant reflux can cause cancer of the sphincter.<QUS> Is this true for people with GERD?*
Non-question statement	*I think that's the key. I'm not sure what you mean by "3 components". The rotation is defined by the third component. The fourth component is used to determine the degree of rotation. So, if you have a 3d vector, you can rotate it in any direction. But if you want to rotate it out to a 4d vector you need to use a fourth component.*
Excessive context duplication	*Now try doing it for tomorrow's price chart. When you look at the historical chart of a company's price you'll be able to pick good buying and selling points 100% of the time. Now try to do it for today's price* Is that impossible?

Table 18: Follow-up Question Error Types and Examples.

F Model Evaluation - Human Annotation Guideline

Table 19 presents the job description and annotation questions for our human annotation task.

Job Description
In this job, you'll be helping us evaluate the quality of follow-up questions generated by a language model called BART. For each task, we will provide you with a pair consisting of a question and answer collected from Reddit's "Explain Like I'm Five" (ELI5) forum. You will be asked to evaluate the quality of the follow-up question generated by BART. These questions and answers aim to provide layperson-friendly explanations for real-life queries. Our data may contain noise, such as invalid follow-up questions, errors, lack of reasoning, or follow-up questions unrelated to the original question or answer. Your role is to help us identify these noisy samples. For each task, you will be shown one triple (question, answer, follow-up question). Carefully review each component and answer the following questions based on your judgment:
Q1: Do you think the follow-up question is a valid question? **A.** Yes **B.** No
Q2: How relevant is the follow-up question to the original question and answer? **A.** Strongly Related **B.** Related **C.** Slightly Related **D.** Not Related
Q3: Does the follow-up question contain any of the following errors? **A.** No Errors **B.** Redundant **C.** Repetitive **D.** Wrong Semantic Collocation **E.** Other Errors
Q4: Does generating this follow-up question require reasoning? **A.** Requires complex amount of reasoning **B.** Requires moderate amount of reasoning **C.** Requires minimal amount of reasoning **D.** Does not require any reasoning
Q5: Does the follow-up question contain new information for the audience? **A.** Introduces a lot of new information **B.** Introduces some new information **C.** Introduces little new information **D.** Does not introduce any new information

Table 19: Task description and evaluation questions used for BART follow-up question evaluation.

F.1 Error Question Guideline

Does the follow-up question contain any of the following errors?

Identify any language issues in the follow-up question.

- **No Errors** – The follow-up question is appropriate and adds value.
- **Redundant** – The follow-up does not introduce any new information.
- **Repetitive** – The follow-up question closely mirrors the original question.
- **Wrong Semantic Collocation** – The question contains unnatural or incorrect phrasing.
- **Other Errors** – Any issues that do not fit the categories above.

Table 20 contains examples of follow-up questions with various error status.

Initial Question: How do vaccines work?	
Initial Answer: Vaccines work by training your immune system to recognize and fight specific germs. They contain harmless parts of the germ (or something similar) so that your body can learn to defend against it. This way, if you ever encounter the actual germ, your immune system can respond quickly and prevent illness.	
No Errors Example	**Redundant Example**
How does a vaccine train the immune system?	Are vaccines used to help the immune system recognize germs?
Reason	**Reason**
The follow-up question is well-formed, relevant, and adds value by diving deeper into a key concept from the original answer. It does not repeat information unnecessarily or contain any language errors.	The follow-up question is redundant because it merely restates information already provided in the initial answer without adding depth or prompting new discussion.
Repetitive Example	**Wrong Semantic Collocation Example**
What do vaccines do?	Do vaccines memorize diseases?
Reason	**Reason**
This follow-up question is nearly identical to the original question, simply reworded. Since it does not introduce new angles or expand on any details, it is considered repetitive.	The phrase"vaccines memorize diseases" is unnatural and incorrect in this context. A better way to phrase the question would be: "Do vaccines help the immune system remember diseases?"

Table 20: Examples of follow-up questions' error status for the given initial question and answer, accompanied by corresponding reasonings below.

F.2 Reasoning Question Guideline

Evaluate the level of reasoning needed to generate the follow-up question.

- **Complex reasoning** involves synthesizing multiple ideas or deeply analyzing information.
- **Moderate reasoning** requires interpreting the given content or slightly extending the discussion.
- **Minimal reasoning** involves simple comprehension or directly rephrasing information.
- **No reasoning** applies to questions that are direct repetitions or restatements without any thought process.

Table 21 contains examples of follow-up questions with various reasoning complexity.

Initial Question: How does sleep affect brain function?	
Initial Answer: Sleep is essential for brain function because it helps with memory consolidation, cognitive processing, and emotional regulation. During sleep, the brain strengthens neural connections, removes toxins, and allows different areas to reset for the next day.	
Complex Amount of Reasoning Example	**Moderate Amount of Reasoning Example**
What are the long-term cognitive effects of chronic sleep deprivation compared to occasional sleep loss?	How does sleep remove toxins from the brain?
Reason	**Reason**
This follow-up question requires complex reasoning because it involves comparing two different scenarios (chronic vs. occasional sleep deprivation) and analyzing their distinct long-term effects on cognition, requiring deeper thought and synthesis of information.	This follow-up question requires moderate reasoning because it builds on a specific detail from the original answer (toxin removal) and asks for an explanation of the biological process involved.
Minimal Amount of Reasoning Example	**Does Not Require Any Reasoning Example**
What are the benefits of sleep for memory?	Does sleep help with memory?
Reason	**Reason**
This follow-up question requires minimal reasoning as it only asks for elaboration on a topic already stated in the original answer (memory consolidation), without introducing any new angle.	This follow-up question does not require any reasoning since it directly repeats a fact already stated in the original answer, making it redundant.

Table 21: Examples of follow-up questions' reasoning complexity for the given initial question and answer, accompanied by corresponding reasons below.

F.3 Informativeness Question Guideline

Evaluate whether the follow-up question enriches the topic by providing or eliciting new information.

- **A Lot of New Information** indicates a significant amount of new knowledge is introduced.
- **Some New Information** suggests moderate enrichment.
- **Little New Information** implies minimal addition.
- **No New Information** means no new information is provided to the audience.

Table 22 contains examples of follow-up questions with various informativeness levels.

Initial Question: How do vaccines work?	
Initial Answer: Vaccines train the immune system to recognize and fight specific germs by introducing harmless parts of the germ or something similar. This prepares the body to respond quickly if exposed to the actual germ in the future.	
A Lot of New Information Example	**Some New Information Example**
What are the differences between traditional vaccines and mRNA vaccines?	How long does it take for a vaccine to provide immunity?
Reason	**Reason**
This follow-up question introduces a significantly new dimension by asking about different types of vaccines, which were not mentioned in the original answer, expanding the discussion substantially.	The follow-up question adds moderately new information by focusing on the timeline of immunity development, a relevant but additional detail not covered in the initial answer.
Little New Information Example	**Does Not Introduce Any New Information Example**
Do vaccines help prevent disease outbreaks?	Do vaccines help the immune system recognize germs?
Reason	**Reason**
The follow-up question slightly expands the discussion by addressing disease outbreaks, but it is already implied in the original answer, as vaccines train the immune system to fight germs.	This follow-up question does not add any new information as it directly restates a key point from the original answer in slightly different words.

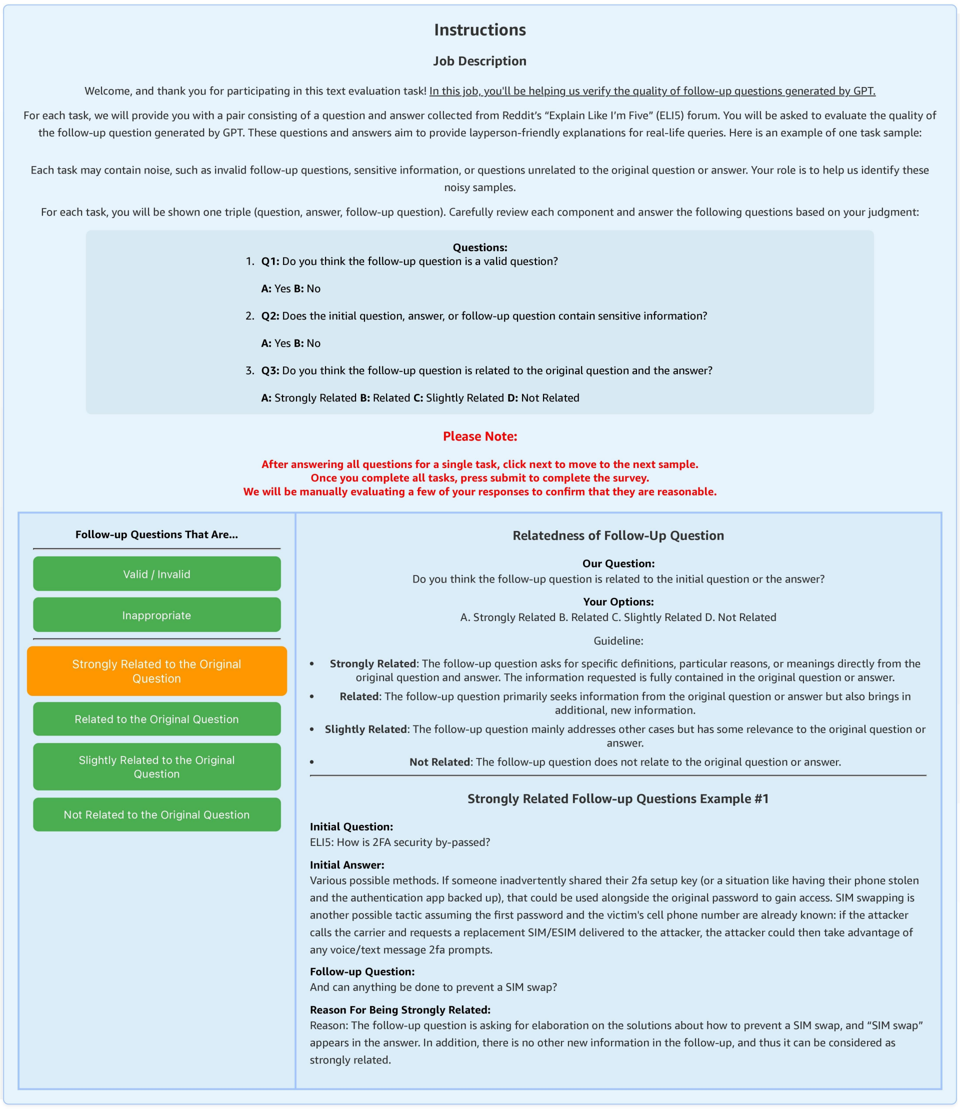
Table 22: Examples of follow-up questions' informativeness for the given initial question and answer, accompanied by corresponding reasons below.

G Additional Examples

See Tables 23

H Interface Examples

See Figures 4 and 5



Instructions

Job Description

Welcome, and thank you for participating in this text evaluation task! In this job, you'll be helping us verify the quality of follow-up questions generated by GPT.

For each task, we will provide you with a pair consisting of a question and answer collected from Reddit's "Explain Like I'm Five" (ELI5) forum. You will be asked to evaluate the quality of the follow-up question generated by GPT. These questions and answers aim to provide layperson-friendly explanations for real-life queries. Here is an example of one task sample:

Each task may contain noise, such as invalid follow-up questions, sensitive information, or questions unrelated to the original question or answer. Your role is to help us identify these noisy samples.

For each task, you will be shown one triple (question, answer, follow-up question). Carefully review each component and answer the following questions based on your judgment:

Questions:

1. **Q1:** Do you think the follow-up question is a valid question?

   A: Yes **B:** No

2. **Q2:** Does the initial question, answer, or follow-up question contain sensitive information?

   A: Yes **B:** No

3. **Q3:** Do you think the follow-up question is related to the original question and the answer?

   A: Strongly Related **B:** Related **C:** Slightly Related **D:** Not Related

Please Note:

After answering all questions for a single task, click next to move to the next sample.
Once you complete all tasks, press submit to complete the survey.
We will be manually evaluating a few of your responses to confirm that they are reasonable.

Follow-up Questions That Are...
Valid / Invalid
Inappropriate
Strongly Related to the Original Question
Related to the Original Question
Slightly Related to the Original Question
Not Related to the Original Question

Relatedness of Follow-Up Question

Our Question:
Do you think the follow-up question is related to the initial question or the answer?

Your Options:
A. Strongly Related B. Related C. Slightly Related D. Not Related

Guideline:

- **Strongly Related**: The follow-up question asks for specific definitions, particular reasons, or meanings directly from the original question and answer. The information requested is fully contained in the original question or answer.
- **Related**: The follow-up question primarily seeks information from the original question or answer but also brings in additional, new information.
- **Slightly Related**: The follow-up question mainly addresses other cases but has some relevance to the original question or answer.
- **Not Related**: The follow-up question does not relate to the original question or answer.

Strongly Related Follow-up Questions Example #1

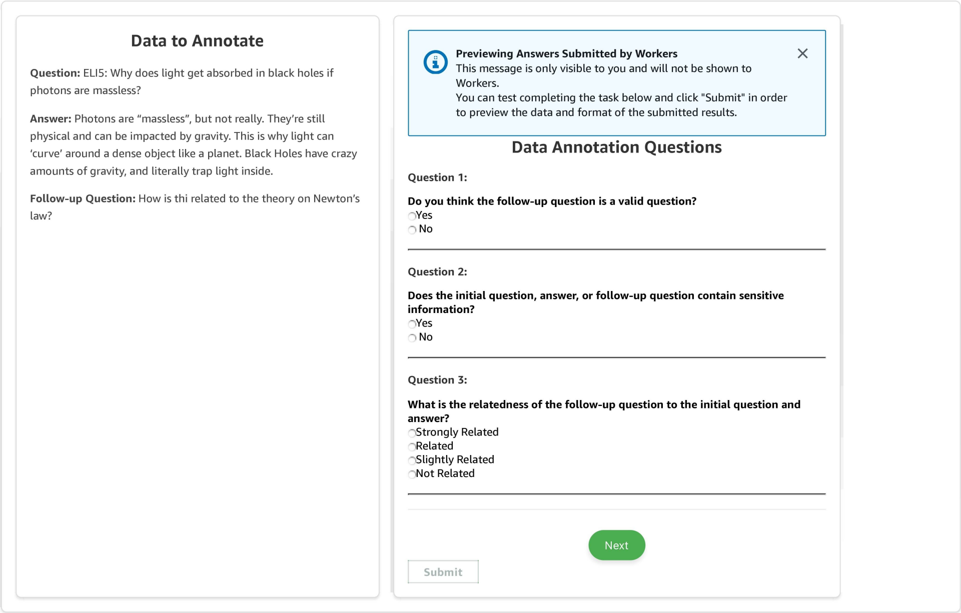
Initial Question:
ELI5: How is 2FA security by-passed?

Initial Answer:
Various possible methods. If someone inadvertently shared their 2fa setup key (or a situation like having their phone stolen and the authentication app backed up), that could be used alongside the original password to gain access. SIM swapping is another possible tactic assuming the first password and the victim's cell phone number are already known: if the attacker calls the carrier and requests a replacement SIM/ESIM delivered to the attacker, the attacker could then take advantage of any voice/text message 2fa prompts.

Follow-up Question:
And can anything be done to prevent a SIM swap?

Reason For Being Strongly Related:
Reason: The follow-up question is asking for elaboration on the solutions about how to prevent a SIM swap, and "SIM swap" appears in the answer. In addition, there is no other new information in the follow-up, and thus it can be considered as strongly related.

Figure 4: Human Evaluation Interface - Task Instructions and Examples.



Data to Annotate

Question: ELI5: Why does light get absorbed in black holes if photons are massless?

Answer: Photons are "massless", but not really. They're still physical and can be impacted by gravity. This is why light can 'curve' around a dense object like a planet. Black Holes have crazy amounts of gravity, and literally trap light inside.

Follow-up Question: How is thi related to the theory on Newton's law?

Previewing Answers Submitted by Workers
This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.

Data Annotation Questions

Question 1:

Do you think the follow-up question is a valid question?
○Yes
○No

Question 2:

Does the initial question, answer, or follow-up question contain sensitive information?
○Yes
○No

Question 3:

What is the relatedness of the follow-up question to the initial question and answer?
○Strongly Related
○Related
○Slightly Related
○Not Related

Figure 5: Human Evaluation Interface - Annotation.

IQ	What causes occupational domination?
IA	*In IT it is cultural. Women in the U.S. drop out of IT classes because they do not like the male-dominated culture.* Whereas India, Vietnam, and other countries where we outsource our IT have a much higher percentage of female programmers. *It is not something about women.* It is something about western IT classes and tech culture.
CA	Occupation domination occurs due to a combination of historical, social, economic, and geographic factors. *Historically*, certain groups have had greater access to education, resources, and opportunities, leading them to be more prevalent in specific jobs or industries. This is often reinforced by network effects, where individuals within a profession prefer to work with or hire people who share similar backgrounds, cultural norms, or experiences, thus perpetuating the concentration of certain demographics in those fields. *Legal and institutional* barriers can also play a significant role. Discriminatory policies or practices can systematically exclude certain groups from specific professions, allowing other groups, who do not face such barriers, to dominate those fields. *Geographic and environmental* factors further influence occupation domination. The presence of certain natural resources or particular climate conditions in an area can dictate the types of industries that develop there. Local populations often specialize in these industries, passing down specialized skills and knowledge through generations, which solidifies their dominance in those fields over time.
FQ by *ORG*	– But why do women drop out of IT classes? – *In IT it is cultural. Women in the U.S. drop out of IT classes because they do not like the male-dominated culture.* This is a myth. *It is not something about women.* It's a myth that women drop out because of the male-dominant culture. – *So it's cultural?* – What about women who want to be in tech but are not in the workforce? – Is there a cultural component of IT that makes it more difficult for women to break into it?
FQ by *AUG*	– What are some examples of how outsourcing has impacted the gender imbalance in IT? – How do cultural expectations influence the recruitment of female programmers? – Can you explain the concept of 'cultural fit' in the context of occupational domination?
FQ by *FULL*	– How do traditional gender roles in professions like medicine and engineering contribute to occupation domination? – Why might women feel more comfortable pursuing professions outside of traditional professions? – Can you explain the concept of 'perceived value' in the context of occupation domination?

Table 23: Example of follow-up question generated by three model variants, with comprehensive answers (ID 3168).

Injecting Frame Semantics into Large Language Models via Prompt-Based Fine-Tuning

Shahid Iqbal Rai **Danilo Croce** **Roberto Basili**
Department of Enterprise Engineering
University of Rome Tor Vergata, 00133, Rome, Italy
`rjshahidrai@gmail.com` `{croce,basili}@info.uniroma2.it`

Abstract

Large Language Models (LLMs) have demonstrated remarkable generalization across diverse NLP tasks, yet they often produce outputs lacking semantic coherence due to insufficient grounding in structured linguistic knowledge. This paper proposes a novel method for injecting Frame Semantics into a pretrained LLaMA model using Low-Rank Adaptation (LoRA). Leveraging FrameNet (a rich resource of over 1,000 semantic frames) we construct a training corpus comprising structured triples of frame definitions, frame elements, and lexical units. Our method encodes these examples into the model via LoRA adapters and evaluates performance using zero-shot prompting for textual entailment and semantic role labeling (SRL) over FrameNet. Experimental results show that our adapted frame-aware LLM substantially outperforms the baseline across closed, open-ended, and multiple-choice prompts. Moreover, we observe significant improvements in SRL accuracy, demonstrating the efficacy of combining frame-semantic theory with parameter-efficient pretraining.

1 Introduction

Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023) and LLaMA (Dubey et al., 2024) have demonstrated impressive capabilities across a wide range of natural language processing (NLP) tasks. However, despite their generalization strength, these models often lack explicit grounding in linguistic theories, which can occasionally result in fluent outputs that overlook deeper semantic distinctions and, in some cases, lead to factual inconsistencies or semantic hallucinations (Ji et al., 2023). To address this gap, enriching LLMs with structured linguistic knowledge could certainly be beneficial, as improved interpretability may support more reliable and semantically coherent outputs.

One promising direction is the integration of Frame Semantics (Fillmore, 1976), a linguistic theory that connects word semantics to situational, i.e. conceptualized, information in terms of *frames*. Each frame consists of a situation (i.e. the frame) and prototypical participants, known as *Frame Elements* (FEs). It is triggered by specific *Lexical Units* (LUs) in the text. For example, the verb *provide* triggers a SUPPLY frame, with roles such as SUPPLIER, RECIPIENT, and THEME. Unlike purely distributional approaches, Frame Semantics imposes situational constraints on semantic role assignments, grounding language interpretation in real-world scenarios. FrameNet (Baker et al., 1998), a computational resource based on Frame Semantics, offers a comprehensive repository of over 1,000 frames and their annotated instances. In fact, equipping models with frame-level information can make a tangible difference for tasks like semantic role labeling (Das and Smith, 2010), question answering (Madabushi et al., 2024), and even commonsense reasoning (Botschen et al., 2018b; Wang et al., 2021b).

In this work, we present a parameter-efficient method to inject Frame Semantics into large language models (LLMs) through fine-tuning with Low-Rank Adaptation (LoRA) (Hu et al., 2022). Our central idea is to make an abstract linguistic theory usable by LLMs by textualizing its core concepts: we systematically convert FrameNet's structured knowledge (frame definitions, frame elements or FEs, and lexical units or LUs) into natural language examples in the form of question–answer pairs. For instance, we generate prompts that ask for the definition of a frame, the roles it involves, or the words that evoke it, thus producing an artificial dataset that "translates" theoretical content into a format suitable for instruction-based adaptation. In total, this process yields a dataset of 6,628 question–answer pairs covering 60 FrameNet frames. By fine-tuning LLaMA models on this textualized

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 31–47
November 8-9, 2025 ©2025 Association for Computational Linguistics

dataset, we aim to encourage the model to internalize frame-semantic structures and relationships, enabling it to better reason about frames, roles, and their instantiations in text, even in the absence of explicit annotation.

A critical question, however, is whether LLMs fine-tuned on such examples merely *memorize* specific facts about the frames encountered during training, or whether they actually *generalize* frame-semantic knowledge to novel, previously unseen frames. To address this, we explicitly evaluate model performance on both **seen frames** (included in fine-tuning) and **unseen frames** (held out from training). This experimental design allows us to disentangle the model's ability to recall injected knowledge from its capacity to abstract and apply frame-semantic principles to new scenarios-an essential property for robust knowledge integration.

We assess the effectiveness of our approach in two ways. First, we probe the model's frame-semantic competence by evaluating its ability to answer structured questions about frames, elements, and lexical units-essentially measuring whether the injected knowledge is accessible via prompting. Second, and more crucially, we test whether this knowledge generalizes to downstream tasks for which the model has not seen explicit training examples. In particular, we consider semantic role labeling (SRL): given a sentence, can the model correctly identify and assign core frame elements? Notably, during fine-tuning, the model is never shown labeled sentences (only definitions and conceptual relations) so improvements on SRL reflect genuine semantic knowledge transfer. Our results show that the frame-aware LLM not only answers frame-related questions more accurately, but also outperforms the baseline on zero-shot SRL tasks, supporting the claim that structured linguistic knowledge can be effectively injected via prompt-based fine-tuning.

Our main contributions are as follows: 1) We propose a lightweight, LoRA-based method for injecting frame-semantic knowledge into LLMs using structured FrameNet annotations. 2) We design a diverse set of instructional prompting templates and linguistic variations to simulate realistic use cases for frame-role understanding. 3) We provide extensive evaluation on both **seen** and **unseen** frames for zero-shot knowledge probing and SRL inference, demonstrating enhanced interpretability and generalization.

In the rest of the paper, Section 2 reviews related work, Section 3 describes our methodology, Section 4 presents experiments and results, and Section 5 concludes with final remarks and future directions.

2 Background and Related Work

Frame Semantics, introduced by Fillmore (Fillmore, 1976), provides a principled approach to modeling linguistic meaning by organizing words into conceptual structures called *frames*. Each frame represents a prototypical scenario, described by a set of frame elements (FEs), and is evoked by specific lexical units (LUs). The FrameNet project (Baker et al., 1998) operationalizes this theory by cataloguing over 1,000 frames, their core and peripheral elements, and annotated instances.

While Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023) and LLaMA (Dubey et al., 2024) achieve remarkable performance across diverse NLP tasks, they are pre-trained on general web corpora and lack explicit integration of structured linguistic resources like FrameNet. As a result, LLMs may generate fluent yet semantically misaligned outputs when required to interpret or generate language in terms of frame-semantic roles.

Recent research has sought to bridge this gap by augmenting LLMs with frame-semantic knowledge. Fine-tuning LLMs on FrameNet data has been shown to enhance their ability to model semantic structures and improve interpretability (Cui and Swayamdipta, 2024a; Torrent et al., 2022). Several works have proposed injecting frame-level information into transformer architectures to support semantic role labeling (Das and Smith, 2010; Zhang et al., 2023), question answering (Madabushi, 2024), commonsense reasoning (Botschen et al., 2018a; Wang et al., 2021a), and even named entity recognition (Alexiev and Casamayor, 2016). Frame-based representations have also been leveraged for more robust and factually grounded summarization (Han et al., 2016; Guan et al., 2021).

Despite these advances, most prior work either leverages FrameNet solely as a source of annotations for supervised tasks or incorporates frame information as static features. In contrast, our approach aims to *internalize* frame-semantic knowledge by textualizing FrameNet diverse knowledge into instructional prompts for LLM adaptation. Furthermore, we explicitly assess the quality of injected knowledge by evaluating the model not just on frames used for adaptation, but also on *unseen* frames held out from training, a perspective rarely addressed in prior studies.

In summary, while previous research has demonstrated the benefits of integrating frame-semantic supervision into neural models, there remains a need for approaches that support robust generalization and interpretability via explicit, structured knowledge injection. Our work aims to reduce this gap by proposing a scalable, prompt-based method for frame-semantic adaptation, and by providing a systematic evaluation on both in-domain and out-of-domain (seen/unseen) frames.

Our work also relates to recent efforts in discourse semantics that employ question answering as a tool for evaluating consistency and logical understanding. For example, (Miao et al., 2024) introduce a Socratic QA framework to test whether LLMs respond consistently to logically equivalent or entailed discourse questions, while (Rabinovich et al., 2023) propose QUDeval to measure semantic consistency across related QA pairs grounded in discourse theory. These studies highlight the importance of consistency in QA-based evaluation, which is complementary to our focus on injecting frame-semantic knowledge into LLMs.

3 Injecting Frame-Semantics into LLMs

Our knowledge injection pipeline, illustrated in Figure 1, is designed to make the structured content of FrameNet directly usable by large language models. The process begins with the extraction of frame-level information from FrameNet: for each frame, we collect its definition (a concise description of the scenario the frame represents), its core frame elements (the prototypical participants or roles involved), and the set of lexical units (words or multiword expressions that evoke that frame in context).

To give a concrete example, Table 1 shows the SUPPLY frame: its definition describes a scenario where a "SUPPLIER provides a THEME to a RECIPIENT." The core frame elements here are roles such as SUPPLIER, RECIPIENT, and THEME, each mapping to a participant in this scenario, for instance, "*China*" as the SUPPLIER, "*Iran*" as RECIPIENT and "*decontamination materials*" as the THEME in the sample sentence. The associated lexical units (LUs) are verbs and nouns like "*provide*", "*supply*", or "*equipment*", each capable of triggering the frame in different contexts.

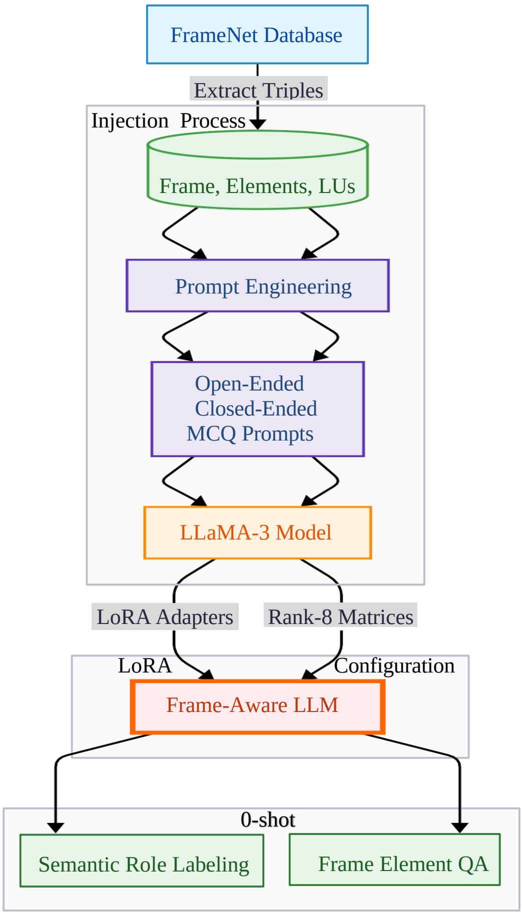


Figure 1: System architecture for frame-semantic knowledge injection into LLaMA-3. The pipeline extracts frame-element-lexical unit triples from FrameNet, converts them into multiple prompt formats (open-ended, closed-ended, MCQ), and fine-tunes the model using LoRA adapters. The resulting Frame-Aware LLM enables zero-shot semantic role labeling and frame element QA.

Frame: SUPPLY		
A SUPPLIER provides a THEME to a RECIPIENT		
FEs	SUPPLIER	Has **China** *provided* Iran with decontamination materials?
	RECIPIENT	Has China *provided* **Iran** with decontamination materials?
	THEME	Has China *provided* Iran **with decontamination materials**?
LUs	*afford.v, equip.v, equipment.n, fix up.v, fuel.v, furnish.v, issue.v, outfit.v, provide.v, provision.n, provision.v, supplier.n, supply.n, supply.v*	

Table 1: Illustration of the SUPPLY frame with annotated frame elements and associated lexical units.

Rather than using FrameNet only as a source for supervised labeling, we transform this structured knowledge into a set of natural language question–answer pairs. For each frame, we generate prompts that ask about its definition, the roles it contains, or which words evoke it, simulating realistic queries a user or downstream application might pose. Prompts are generated in various formats, including open-ended, closed-ended, and multiple-choice ensuring broad coverage of the theory.

This synthetic QA dataset serves as supervision for fine-tuning a pretrained LLaMA-3 model via Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA is a parameter-efficient fine-tuning technique that augments a frozen pretrained model with small trainable low-rank matrices. During training, only these additional parameters are updated, greatly reducing memory and computational cost while preserving the general linguistic competence already encoded in the model. These properties make LoRA particularly suitable for injecting structured resources like FrameNet, where the large number of frame–role combinations would make full fine-tuning both expensive and prone to catastrophic forgetting, i.e., the overwriting of previously acquired knowledge. By constraining learning to a compact set of additional parameters, LoRA enables the integration of frame-semantic knowledge without erasing the model's broader abilities. In our experiments, we adopt both 3B and 8B LLaMA models, which balance computational feasibility with meaningful evaluation.

This allows the model to internalize the relationships between frame definitions, roles, and lexical units, without relying on explicit sentence-level annotation. After training, the resulting Frame-Aware LLM can be probed on zero-shot tasks such as semantic role labeling and frame-related question answering.

Instructional Template Construction. To effectively inject frame-semantic knowledge into large language models (LLMs), we design natural language templates (Zheng et al., 2023; Su et al., 2021; Wen et al., 2024) that translate structured FrameNet annotations-such as frame definitions, frame elements (FEs), and lexical units (LUs)-into instructive, contextualized prompts. In the *Question: "Can you list some frame elements in the X frame?"* with *Answer: "The frame elements of the X frame are:* FE_1*,* FE_2*, and* FE_3." symbols, such as X or FE_1, are placeholders replaced with annotations from FrameNet. This approach builds on the principle that linguistic structure can be aligned with QA-based representations (He et al., 2015), supporting both training as well as augmenting interpretability.

We compose 11 task-specific templates, grouped as follows: six open-ended, four closed-ended, and one multiple-choice (MCQ) format. Each template addresses a distinct aspect of frame-semantics, ranging from recognizing frame elements to identifying lexical units and mapping roles to frames. This diversity enables the model to encounter a wide range of linguistic formulations, enhancing generalization (Ma et al., 2022; Cui and Swayamdipta, 2024b).

Prompt Types:

- **Open-ended Prompts** (6 templates): Encourage free-form, descriptive responses and probe the model's ability to verbalize frame knowledge in its own words. These cover frame definitions (e.g., *What is the definition of the X frame?*), frame elements (e.g., *Can you list some frame elements in the X frame?*), frame element definitions, and lexical units.
  Example: Question: Can you identify a few frame elements or roles in the 'SUPPLY' *frame?*
  *Answer: The frame elements "*SUPPLIER, THEME *and* RECIPIENT*" are associated with the* SUPPLY *frame.*

- **Closed-ended Prompts** (4 templates): Binary (yes/no) or direct verification questions to check specific facts about frames, roles, or lexical units.
  Example: Question: Are the roles 'RECIPIENT' *and* 'THEME' *part of the frame elements of the* 'SUPPLY' *frame?*
  Answer: Yes

- **Multiple-choice Prompts** (1 template): The model selects the correct answer among several options, diagnosing confusion or gaps in understanding.
  Example: Question: Which role is part of the frame elements in the 'SUPPLY' *frame?*
  A) SUPPLIER *B)* RECIPIENT *C)* LOCATION *D)* THEME
  Answer: D) THEME

Linguistic Variations. Template diversity alone is not sufficient to guarantee robustness: a model could simply memorize fixed associations between question forms and answers. To promote generalization, for each template and frame, we systematically construct five alternative phrasings of the question and fifteen variants of the answer. For instance, for the "list frame elements" template, questions might include: *Which are the roles in the X frame?*, *What are some frame elements defined for X?*, *Who are the core entities in the X frame?*, etc. Answers likewise vary (e.g., *The X frame includes* FE_1, FE_2, *and* FE_3; FE_1, FE_2, *and* FE_3 *define the X frame*; and so on).

For each training instance, one question and one answer variant are chosen at random and the pair is used as a supervised example. This strategy, inspired by Dong et al. (Dong et al., 2017), exposes the model to a wide spectrum of natural language formulations and minimizes spurious correlations, crucial for supporting transfer to unseen frames (see Section 4). In this way, the LLM cannot "cheat" by matching surface forms; it must internalize the underlying frame-semantic associations. Taken together, our prompt engineering pipeline, spanning diverse task templates and systematic linguistic variation, supports both the depth and breadth of frame-semantic knowledge acquisition. This methodology improves robustness, interpretability, and aligns the knowledge injection process more closely with the real-world variability of language. To further ensure quality, we manually inspected around 100 generated SRL examples, confirming that the questions and answers were consistent with the intended frame-semantic annotations (Mihaylov et al., 2018).

4 Experimental Evaluation

In this section, we evaluate the effectiveness of our frame-semantic knowledge injection approach for large language models (LLMs). Our experimental objectives are twofold: (1) determine whether the injected knowledge substantially enhances the model's ability to reason about frames, frame elements, and lexical units; and (2) assess whether this acquired semantic knowledge generalizes effectively to practical downstream tasks, most notably, semantic role labeling (SRL)-even without explicit SRL supervision during training.

4.1 Experimental Setup

Our experiments utilize FrameNet version 1.7[1] (Baker et al., 1998), a comprehensive lexical database cataloging over 1,000 semantic frames, their associated core and peripheral frame elements, and lexical units.

Frame Selection and Dataset Composition. For our initial evaluation, we constructed a representative subset of 60 frames from FrameNet, designed to maximize semantic diversity and ensure robust hierarchical coverage (see Appendix A for full criteria and the frame list). The selection process began with a set of core "seed" frames (such as ABANDONMENT, BRINGING, ASSISTANCE, MOTION, and COMMUNICATION) which were chosen to span different domains and frame complexities. From these seeds, we expanded the set by systematically including frames that are hierarchically related, either *inheriting from* or *being inherited by* others within the FrameNet taxonomy. This relational expansion yielded a set of frames that are both semantically coherent and structurally interconnected, capturing the full breadth of frame–element–lexical unit configurations observed in FrameNet. As a result, the final subset covers 175 unique frame elements and 730 lexical units, with frames selected to reflect a broad range of structures (from simple to highly articulated) and to ensure that all major types of frame–element relations and domains are represented. This principled construction ensures the resulting dataset is both challenging and realistic for frame-semantic evaluation.

Instance Sampling Strategy. Naturally, the sampling process differs slightly depending on the type of task. For now, we disregard the additional layer of linguistic variation and focus on the core instance generation procedure. For every open-ended task, the approach is straightforward: for example, when eliciting the definition of a frame or asking which frame corresponds to a given definition, a single core instance is generated per frame (before further expansion via linguistic paraphrasing). However, for tasks that involve frame elements (restricted here to core frame elements) or lexical units, the number of instances per frame directly depends on the number of relevant elements or units present in that frame. In other words, frames with more core frame elements or lexical units will yield proportionally more question–answer pairs for those tasks. Further details on the sampling strategies adopted for frame elements and lexical units are provided in Appendix B. Closed-ended tasks require both positive and negative examples to prevent the model from defaulting to trivial responses (e.g., always answering *no*). Positive samples are created by pairing correct annotations (e.g.,

---

[1] `https://framenet.icsi.berkeley.edu/frameIndex`

frame definitions, frame elements) with their respective frames. Negative samples, however, must be selected carefully to avoid class imbalance: using all incorrect annotations would overwhelm the dataset with negatives. To address this, we fix the ratio to $p = 3$ positives and $q = 6$ negatives per frame–task pair. Negatives are drawn from unrelated frames and filtered to avoid duplication. This results in a balanced and informative training signal: $p - 2$ positive and $q - 2$ negative samples are assigned to training, with the rest evenly split across validation and test. Full sampling procedures for specific tasks are detailed in Appendix C. Multiple-choice tasks were constructed with a fixed number of $k = 5$ samples per frame, allocated as $k - 2 = 3$ to the training set, and one each to the validation and test sets. Each MCQ instance presented a single correct answer along with a set of distractors sampled from alternative frames, ensuring that all options were unique and plausible. To increase the challenge and diagnostic value, some prompts included an additional distractor option such as "None of these", following practices proposed in prior work (Yatskar et al., 2016). The training split intentionally contained both positive and fully negative MCQs (i.e., questions with only incorrect options), while the validation and test sets each included one positive and one negative sample per frame to support balanced evaluation.

Synthetic Dataset Construction. From the targeted frames, we systematically generated a total of 6,628 synthetic question–answer pairs, employing linguistically diverse prompt templates (described in Section 3 and exemplified in Appendix D). Following generation, we allocated 3,642 samples to the training set, 1,493 to the development (validation) set, and 1,493 to the test set. Except this split, with same approach a separate set of 1,052 question–answer pairs was generated using 10 unseen frames to evaluate the model's generalization at unseen frame-semantic knowledge. Moreover, no question or answer surface form is ever repeated across different splits, preventing the model from memorizing fixed linguistic patterns. Each of the 11 tasks was instantiated using multiple paraphrased templates for both questions and answers. Specifically, for each task, three distinct question formulations were assigned to the training set, one to the validation set, and one to the test set. Answer templates followed a similar logic: out of a total of 15 available variants per task, 10 were designated as eligible for training (from which 3 were randomly sampled for each frame), 2 were allocated for validation (with one randomly selected), and 3 were reserved for testing (with one randomly selected). This controlled partitioning ensures strict paraphrastic separation across splits, preventing the model from relying on surface-form memorization and encouraging genuine generalization. Concrete examples and the complete set of question–answer paraphrases for a representative frame-based task are provided in Appendix D. In particular, a detailed summary of the prompt types, task formulation strategies, and sampling counts is reported in Table 6.

Evaluating Generalization. To rigorously assess the model's ability to generalize beyond memorization, we adopted a frame-level splitting strategy rather than random sampling: 50 frames were designated as *seen* (utilized for training and validation), while 10 frames were held out as *unseen* and reserved exclusively for zero-shot evaluation. The unseen frames, such as RELEASING, MANIPULATION or CONTROL, were selected to ensure semantic and structural diversity against phenomena not observed during training. Further details on the selection process, as well as the full frame list and distribution across tasks, are provided in Appendix A.

Fine-tuning Configuration. We fine-tuned pretrained LLaMA models-LLaMA 3.2 3B[2] and LLaMA 3.1 8B[3], using Low-Rank Adaptation (LoRA) (Hu et al., 2022), building on the architecture described in (Touvron et al., 2023). LoRA introduces trainable low-rank matrices into the model's attention and feedforward layers, allowing for parameter-efficient adaptation with minimal computational overhead. In our experiments, we used a rank of 16, a scaling factor of $\alpha = 16$, and no dropout. Fine-tuning was carried out using instruction-style prompts consistent with the supervised instruction tuning paradigm (Ouyang et al., 2022). Each input was structured in a standardized format: `### Input: <QUESTION> - ### Response: <ANSWER>`. We employed the Unsloth framework (Daniel Han and team, 2023) to enable efficient fine-tuning with 4-bit quantized weights. Models were trained for 7 epochs using

---

[2] `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`

[3] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

the AdamW optimizer (8-bit), with a learning rate of 2×10^{-4}, a batch size of 4 per device, and 8 gradient accumulation steps. To optimize memory usage, we activated gradient checkpointing and used FP16 or BF16 precision depending on hardware capabilities. Model selection was based on validation loss, evaluated every 50 steps. All fine-tuned LoRA models [4][5] and associated tokenizers have been released on the Hugging Face Hub (Wolf et al., 2020) to facilitate reproducibility.

4.2 Results and discussion

Evaluation of Injected Frame-Semantic Knowledge. To evaluate the effectiveness of our frame-aware supervision strategy, we group the 11 frame-related tasks into three broad categories based on the type of prompt: *Closed-ended*, *Open-ended*, and *Multiple-choice questions (MCQs)*. Each category reflects a different cognitive demand: Closed-ended tasks involve binary decisions (e.g., verifying if a role belongs to a frame), Open-ended tasks require free-form generative responses (e.g., defining a frame or a role), and MCQs present a set of options from which the model must select the correct answer. We further assess performance under two generalization regimes. The first includes the 50 *seen* frames used during training and validation ("In-domain"). The second consists of 10 *unseen* frames ("Out-of-domain"), explicitly held out for zero-shot evaluation to test the model's ability to generalize beyond the training distribution. Results for both the baseline LLaMA models (3.1 8B and 3.2 3B) and their fine-tuned variants are summarized in Table 2. Each prompt category is evaluated using a metric suited to its output type. For both Closed-ended and Multiple-choice (MCQ) tasks, we report the F1 score, which balances precision and recall, effectively capturing the model's ability to make accurate binary and categorical predictions. Although MCQs involve a selection among distractors, their scoring is treated as a binary classification of correctness, hence the use of F1. For Open-ended tasks, which require the model to produce free-form natural language responses (e.g., definitions, descriptions of frame elements), we adopt a semantic similarity metric. Specifically, we compute the cosine similarity between the predicted and reference answers using Sentence-BERT embeddings[6]. This approach, standard in semantic textual similarity evaluation, allows us to assess whether the model captures the intended meaning even when surface forms differ.

Fine-tuned models consistently outperform their pretrained counterparts across all prompt types and model sizes. For instance, the LLaMA 3.1 8B model shows a substantial improvement in F1 score on Closed prompts, rising from 0.55 to 0.93, and in cosine similarity on Open-ended tasks, from 0.64 to 0.87. The gains extend to Multiple-choice questions as well, with F1 increasing from 0.27 to 0.66. The smaller LLaMA 3.2 3B model exhibits similar trends, confirming the robustness of the approach. These results demonstrate that injecting structured frame-semantic supervision significantly enhances the model's ability to understand and reason over semantic roles, definitions, and frame-element associations. Despite not being exposed to ten frames during training, fine-tuned models retain strong performance on these held-out examples. For instance, the 8B model drops only slightly from 0.87 to 0.73 in cosine similarity on Open-ended prompts, and from 0.93 to 0.87 in F1 on Closed ones. This small degradation indicates that the model generalizes well beyond memorization, applying abstract frame-semantic reasoning to novel lexical and conceptual configurations. The original LLaMA models perform consistently worse in all settings. On unseen frames, the baseline 3.2 3B model achieves only 0.33 F1 on MCQs and 0.63 cosine on Open-ended tasks. In contrast, the fine-tuned models maintain substantially higher scores. These discrepancies highlight the necessity of targeted, frame-aware training signals: without them, the model struggles to interpret even well-formed prompts about roles, definitions, or lexical associations. In sum, our results provide strong evidence that explicit semantic supervision-grounded in FrameNet and enhanced by task- and template-level diversity-substantially improves the model's ability to understand and manipulate frame-semantic knowledge. Notably, the generalization observed on zero-shot frames suggests that the learned representations are not only effective but also transferable, paving the way for broader deployment in downstream tasks such as frame disambiguation, SRL, and knowledge-based QA.

---

[4] https://huggingface.co/shahidrai/llama_3.1_8b/tree/main

[5] https://huggingface.co/shahidrai/llama_3.2_3b_finetuned/tree/main

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

A detailed breakdown of performance across individual tasks is provided in Appendix E.

Model	Pr. Type	Metric	Zero-shot		Fine-tuned	
			In	Out	In	Out
LLaMA 3.1 8B	Closed	F1	0.55	0.52	**0.93**	**0.87**
	Open-ended	Cos	0.64	0.63	**0.87**	**0.73**
	MCQs	F1	0.27	0.34	**0.66**	**0.63**
LLaMA 3.2 3B	Closed	F1	0.49	0.50	**0.91**	**0.86**
	Open-ended	Cos	0.63	0.63	**0.83**	**0.73**
	MCQs	F1	0.25	0.33	**0.52**	**0.50**

Table 2: Performance comparison across models and prompt types, grouped by evaluation setting. "In" refers to seen frames; "Out" refers to unseen frames.

Semantic Role Labeling (SRL) Evaluation. To evaluate whether frame-semantic knowledge acquired through our supervision strategy transfers to a practical downstream task, we design a controlled Semantic Role Labeling (SRL) experiment. In this setting, the model is prompted to identify the lexical unit evoking a given frame and extract the associated frame elements expressed in a sentence. We employ a zero-shot prompting strategy inspired by instructional paradigms (Devasier et al., 2025), using structured, natural language instructions (detailed in Appendix 2) rather than fine-tuning on SRL-annotated data. We evaluate on the Open-Sesame dataset[7] (Swayamdipta et al., 2017), which is based on FrameNet and originally released in CoNLL format (Carreras and Màrquez, 2005). The evaluation set contains 371 sentences covering 38 of the 50 frames used during training, totaling 468 annotated instances. While the task setup is not intended to compete with dedicated SRL systems, it provides a diagnostic test bed to verify whether the model can apply definitional and structural knowledge to recognize semantic roles in naturalistic text.

Table 3 breaks down model performance across three increasingly strict evaluation criteria for frame element identification. The first row (Roles Only) considers predictions correct if the role label matches, regardless of span alignment. The second criterion (Roles + Span (25%)) adds a minimum 25% token-level overlap requirement between the predicted and gold spans. The final setting (Roles + Span (75%)) requires a much tighter alignment, with at least 75% span overlap. The fine-tuned model significantly outperforms the baseline in all settings, achieving a fourfold improvement in role-only detection (0.60 vs. 0.14) and similarly large gains in span-aware scoring (e.g., 0.41 vs. 0.10 at 25% threshold). Even under the strictest criterion (75% overlap), it reaches 0.25 F1, far surpassing the baseline's 0.07.

Consider the sentence: *"Has China provided Iran with decontamination materials?"* In this representative example involving the SUPPLY frame, both the base LLaMA model and our fine-tuned version correctly identify *China* as the SUPPLIER. However, the base model incorrectly labels *Iran* as a LOCATION and fails to detect any additional role. In contrast, our fine-tuned model correctly assigns the RECIPIENT role to *Iran* and identifies *materials* as the THEME. Although the predicted span misses part of the full constituent ("with decontamination materials"), it successfully captures the semantic head, which is often sufficient for downstream tasks. This pattern is consistent with our overall results: the fine-tuned model reliably recovers nearly all core roles, in line with the aggregate metrics, but span completeness can occasionally be imprecise.

While our setup is simplified and intentionally scoped, it is noteworthy that the fine-tuned model achieves competitive (if not superior) performance compared to recent LLM-based SRL systems. For example, Cheng et al. (2024) report F1 scores of 0.40 and 0.38 using ChatGPT in a 3-shot setting on *CoNLL-2005 WSJ* and *CoNLL-2012 WSJ*, respectively, and just 0.22 F1 in a zero-shot setting on *CoNLL-2005 WSJ*. In contrast, our model reaches 0.41 F1 in zero-shot SRL, despite being trained on a smaller and more focused dataset comprising only 50 FrameNet frames. This discrepancy can be explained in part by the underlying resource differences: while Cheng et al. (2024) evaluate over PropBank-style predicates, our approach concentrates on a curated subset of FrameNet frames. This narrower scope likely contributes to the higher accuracy, as it allows the model to internalize more structured and semantically grounded knowledge.

Evaluation Criterion	Zero-shot	Fine-tuned
Roles Only	0.14	**0.60**
Roles + Span (25% ov.)	0.10	**0.41**
Roles + Span (75% ov.)	0.07	**0.25**

Table 3: SRL Performance on Frame Element Prediction (F1 Score)

These results confirm that frame-semantic supervision improves both structural role identification and token-level grounding of semantic roles,

---

[7] `https://github.com/swabhs/open-sesame`

demonstrating generalization from injected knowledge to real-world SRL inputs. These results, although obtained in a controlled setting and limited to a selected subset of frames, mark a promising first step: their consistency indicates that the approach is robust and generalizable. The natural next step is to scale the fine-tuning procedure to the full FrameNet inventory, a conceptually straightforward extension that merely requires a longer training cycle.

A brief analysis highlights both strengths and weaknesses of the fine-tuned model. At the positive end, qualitative analysis shows clear improvements over the baseline. For instance, in the Supply frame the fine-tuned model correctly recovers all gold-standard roles and spans in: "*Has* [*China*]$_{\text{Supplier}}$ *provided* [*Iran*]$_{\text{Recipient}}$ [*with decontamination materials*]$_{\text{Theme}}$*?*". By contrast, the baseline mislabels the Theme span, predicting [*materials*]$_{\text{Material}}$ and missing the full constituent. This suggests that frame semantics help the model align roles and spans more faithfully to gold annotations, correcting systematic errors made by the baseline. At the same time, errors remain. In the Exchange frame, sentence "*The Mycenaeans were an acquisitive race who came to conquer, not to trade*", the fine-tuned model hallucinates Theme and Recipient alongside the correct roles Exchanger 1 and Exchanger 2, inflating false positives. Similarly, in the Objective influence frame, sentence "*Many Jamaicans head to the States for further education, and the American economic influence on areas such as business investment and planning is growing*", it adds a spurious Area role where none was annotated. Another common error is predicting roles without spans, which negatively impacts F1. By contrast, in such difficult cases the baseline typically fails to recover any meaningful roles at all.

5 Conclusion and Observations

In this work, we introduced an efficient and principled methodology for injecting structured frame-semantic knowledge into large language models via LoRA-based fine-tuning. By transforming FrameNet resources into instructional prompts, we enabled the model to internalize rich semantic abstractions grounded in linguistic theory. Our experiments demonstrate substantial gains in both frame and role recognition tasks, as well as in zero-shot semantic role labeling (SRL). Importantly, the model exhibits strong generalization to unseen frames, highlighting its ability to abstract beyond surface-level associations and apply learned structures in novel contexts.

These findings suggest that explicitly aligning LLMs with Frame Semantics can meaningfully enhance their semantic behavior, without sacrificing general language capabilities. This opens promising avenues for future research, including scaling to broader frame inventories, by also exploring more refined prompting strategies. Moreover, we will study the overall impact of the proposed adaptation framework on LLM interpretability and reliability in other downstream tasks, like QA and dialogue. Future work could also explore frame-to-frame relations (e.g., inheritance links), which are highly relevant for reasoning tasks such as NLI where entailment often depends on recognizing hierarchical or causal connections between events. In addition, future evaluations should stratify FrameNet QA data to examine which question types (e.g., frame definitions, frame elements, lexical units) drive the observed improvements, and extend the study across multiple large language models to assess the generalizability of frame-semantic knowledge injection beyond a single architecture.

Acknowledgments

We acknowledge financial support from the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Limitations

This study focuses on a constrained subset of FrameNet frames and tasks, reflecting an intentionally scoped investigation. While our model shows substantial gains in frame-semantic reasoning, several limitations remain. First, it occasionally predicts spurious frame elements, especially in low-resource frames, reducing precision. Second, it often fails to produce accurate spans for correctly identified roles, limiting its effectiveness in span-level SRL. Extending coverage to the full FrameNet inventory and evaluating across additional tasks (e.g., QA, dialogue) are key directions for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Vladimir Alexiev and Gerard Casamayor. 2016. Fn goes nif: integrating framenet in the nlp interchange format. In *Proceedings of the LDL 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources*, pages 1–10.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. Technical Report 98-009, International Computer Science Institute.

Teresa Botschen, Daniil Sorokin, and Iryna Gurevych. 2018a. Frame-and entity-based knowledge for common-sense argumentative reasoning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 90–96.

Theresa Botschen and 1 others. 2018b. Learning to reason with framenet. In *Proceedings of EMNLP*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164.

Ning Cheng, Zhaohui Yan, Ziming Wang, Zhijie Li, Jiaming Yu, Zilong Zheng, Kewei Tu, Jinan Xu, and Wenjuan Han. 2024. Potential and limitations of llms in capturing structured semantics: A case study on srl. *arXiv preprint arXiv:2405.06410*.

Xinyue Cui and Swabha Swayamdipta. 2024a. Annotating framenet via structure-conditioned language generation. *arXiv preprint arXiv:2406.04834*.

Xinyue Cui and Swabha Swayamdipta. 2024b. Annotating FrameNet via structure-conditioned language generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 681–692. Association for Computational Linguistics.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Dipanjan Das and Noah A. Smith. 2010. A probabilistic frame-semantic parser. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956.

Jacob Devasier, Rishabh Mediratta, and Chengkai Li. 2025. Can llms extract frame-semantic arguments? *arXiv preprint arXiv:2502.12516*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 875–886.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250.

Charles J Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021. Frame semantics guided network for abstractive sentence summarization. *Knowledge-Based Systems*, 221:106973.

Xu Han, Tao Lv, Zhirui Hu, Xinyan Wang, and Cong Wang. 2016. Text summarization using framenet-based semantic graph model. *Scientific Programming*, 2016(1):5130603.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774. Association for Computational Linguistics.

Harish Madabushi and 1 others. 2024. Frame-based embeddings for coherent question answering. In *Proceedings of EACL*.

Harish Tayyar Madabushi. 2024. Fs-rag: A frame semantics based approach for improved factual accuracy in large language models. *arXiv preprint arXiv:2406.16167*.

Yisong Miao, Ellie Pavlick, Tom Kwiatkowski, Luke Zettlemoyer, and Pradeep Dasigi. 2024. Discursive socratic questioning: Evaluating the faithfulness of language models' understanding of discourse relations. In *Proceedings of the 62nd Annual Meeting of* *the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6134–6153, Bangkok, Thailand. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ella Rabinovich, Samuel Ackerman, Mo Yu, Kun Xu, Sewon Min, and Dan Roth. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the 3rd Workshop on Generation, Evaluation, and Metrics (GEM) at EMNLP 2023*, pages 119–130, Singapore. Association for Computational Linguistics.

Xuefeng Su, Ru Li, Xiaoli Li, Jeff Z. Pan, Hu Zhang, Qinghua Chai, and Xiaoqi Han. 2021. A knowledge-guided framework for frame identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5230–5240. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *arXiv preprint arXiv:1706.09528*.

Tiago Timponi Torrent, Ely Edison da Silva Matos, Frederico Belcavello, Marcelo Viridiano, Maucha Andrade Gamonal, Alexandre Diniz da Costa, and Mateus Coutinho Marim. 2022. Representing context in framenet: A multidimensional, multimodal approach. *Frontiers in Psychology*, 13:838441.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2714–2730.

Chenhao Wang, Yubo Chen, Zhipeng Xue, Yang Zhou, and Jun Zhao. 2021a. Cognet: Bridging linguistic knowledge, world knowledge and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 18, pages 16114–16116.

Y Wang and 1 others. 2021b. Knowledge-aware commonsense reasoning with framenet. *Unknown*.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering. *arXiv preprint arXiv:2405.14383*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. `https://huggingface.co/docs/transformers`.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rui Zhang, Yajing Sun, Jingyuan Yang, and Wei Peng. 2023. Knowledge-augmented frame semantic parsing with hybrid prompt-tuning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Ce Zheng, Yiming Wang, and Baobao Chang. 2023. Query your model with definitions in FrameNet: An effective method for frame semantic role labeling. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 14029–14037. AAAI Press.

A Task-Frame Sample Distribution

In FrameNet, semantic relationships between frames are organized through hierarchical links, primarily defined as *inherits from* and *inherited by*. To maximize semantic diversity and cover a broad range of frame phenomena, we selected 60 FrameNet frames according to the following principles:

- Each frame maintains at least one hierarchical relationship with another (either *inherits from* or *inherited by*), ensuring structural coverage within the FrameNet taxonomy.
- Selection prioritizes frames with diverse numbers of frame elements and lexical units, and spanning multiple FrameNet domains.
- Ten frames were held out as *unseen* for zero-shot evaluation: RELEASING, MANIPULATION, CONTROL, KIDNAPPING, and COMMUNICATION MANNER (see complete list below in Table 8).

The complete set of frames used for prompt generation and model training is listed in Table 9, where each row represents a FrameNet frame and each column corresponds to a template-based task. The columns of Table 9 are ordered and numbered as follows, and match the templates described in Section 3:

1. **Open-ended:** *What is the definition of the X frame?*
2. **Open-ended:** *Which frame is defined by "def(X)"?*
3. **Closed-ended:** *Is "def(X)" regarded as the definition of Frame X?*
4. **Open-ended:** *Can you list some frame elements in the X frame?*
5. **Open-ended:** *Which is the frame involving frame elements such as FE_1 and FE_2?*
6. **Multiple-choice:** *Which one of the following roles belongs to the set of frame elements of the X frame?*
7. **Closed-ended:** *Are roles such as Y and Z part of the frame elements of Frame X?*
8. **Open-ended:** *How is frame element FEi defined in X frame?*
9. **Closed-ended:** *Does the definition of frame element "def(FE_i)" accurately express FE_i in the X frame?*
10. **Open-ended:** *Could you list some lexical units associated with the X frame?*
11. **Closed-ended:** *Can LU_i, as a POS_i, be considered as the lexical unit of the X frame?*

Each cell in the table reports the number of question–answer pairs generated for the corresponding frame–task combination for details on sampling). The final column reports the total number of samples generated for each frame.

Task Type Details:

- **Open-ended Prompts (Tasks 1, 2, 4, 5, 8, 10):** The model provides free-form or descriptive responses, testing the ability to paraphrase and verbalize frame knowledge.
- **Closed-ended Prompts (Tasks 3, 7, 9, 11):** Require binary (yes/no) or direct verification, probing recognition of frame facts or rejections.
- **Multiple-choice Prompts (Task 6):** The model selects the correct answer among candidates, revealing confusion or gaps.

Unseen Frames for Zero-shot Evaluation: To assess the model's performance after fine-tuning, we apply the same sample generation methodology described in Section 3 to a separate set of unseen frames. By using a uniform prompt structure and evaluation format, we isolate the effect of unseen knowledge on model behavior in a controlled setting. Ten frames Table 8 were excluded from training and validation and used only for zero-shot testing:

- **SUICIDE ATTACK** (inherits from ATTACK)
- **MOTION NOISE** (inherits from MOTION)
- **COMMERCE PAY** (inherits from GIVING)
- **RECEIVING** (inherits from GETTING)
- **CORPORAL PUNISHMENT** (inherits from REWARDS AND PUNISHMENTS)
- **COMMUNICATION MANNER** (inherits from COMMUNICATION)

- **KIDNAPPING** (inherits from COMMITTING CRIME)
- **CONTROL** (inherits from OBJECTIVE INFLUENCE)
- **MANIPULATION** (inherits from INTENTIONALLY ACT)
- **RELEASING** (inherits from INTENTIONALLY AFFECT)

These choices ensure challenging and diverse coverage for generalization evaluation.

B Sampling Strategy for Open-ended Tasks

As described in Appendix A, open-ended tasks are sampled according to the specific structure of each prompt.

For tasks that require only a single reference to the frame-such as asking for the definition of a frame (*"What is the definition of the X frame?"*) or for the frame corresponding to a given definition (*"Which frame is defined by "def(X)"?"*)-the sampling process is straightforward. For each frame, we generate a single instance per prompt type, later augmented through linguistic variation in the main dataset.

In contrast, for open-ended tasks that involve frame elements or lexical units, sampling is more nuanced due to the multiplicity of possible elements within each frame. For example, in prompts like *"Can you list some frame elements in the X frame?"* and *"Which is the frame involving frame elements such as* FE_1 *and* FE_2*?"*, the core frame elements are randomly sampled or grouped, ensuring that not all elements always appear in the same order or configuration.

A particularly important case is the prompt *"How is frame element FE$_i$ defined in the X frame?"*. Here, the number of generated samples is determined by the number of core frame elements associated with each frame. To ensure coverage while controlling dataset size, we generate $p = 3$ variations for each core frame element, allocating $p - 2$ samples to the training set, and one each to the validation and test sets.

For prompts targeting lexical units, such as *"Could you list some lexical units associated with the X frame?"*, we employ the iterative prompting technique from (Wang et al., 2022). The number of samples generated depends on the number of available lexical units, grouped by part of speech (POS; e.g., verbs, nouns, adjectives) within each frame. Specifically, let n_{POS} be the number of lexical units for a given POS. We partition the list of lexical units into sub lists, each containing at most five items. The number of samples s_{POS} for each POS is thus computed as

$$s_{\text{POS}} = \left\lceil \frac{n_{\text{POS}}}{5} \right\rceil$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

For instance, if the frame ARRIVING includes $n_{\text{verb}} = 15$ verb lexical units, then

$$s_{\text{verb}} = \left\lceil \frac{15}{5} \right\rceil = 3$$

resulting in three samples for verbs. If there are $n_{\text{noun}} = 8$ noun lexical units, then

$$s_{\text{noun}} = \left\lceil \frac{8}{5} \right\rceil = 2$$

so two noun samples are produced.

For evaluation, the complete set of lexical units per part of speech is exhaustively covered, with three sample sets (each reflecting a different linguistic variation) distributed across training, validation, and test splits. This guarantees that all lexical units are sampled without repetition or overlap between sets (Fillmore et al., 2003).

C Sampling Strategy for Closed-ended Tasks

Closed-ended tasks require a balanced set of positive and negative examples to support meaningful learning and avoid degenerate behaviors (e.g., always predicting *no*). For tasks such as *"Does the definition of frame element def(FE$_i$) accurately express FE$_i$ in the X frame?"*, we generate $k = p + q = 9$ samples per frame element FE$_i$, where $p = 3$ are positive and $q = 6$ are negative.

Negative examples are created by pairing the target frame with distractor definitions, elements, or lexical units sampled from unrelated frames (e.g., using definitions from ATTACK when evaluating GESTURE). This ensures diversity while avoiding overwhelming the model with negatives. To preserve task balance, $p - 2$ positive and $q - 2$ negative samples are included in the training set, with the remaining examples evenly split between validation and test.

For tasks involving lexical unit verification, such as *"Can LU$_i$, as a POS$_i$, be considered a lexical* *unit of the X frame?"*, samples are generated for each sublist of five lexical units. Given a part of speech POS_i, the number of such sublists is computed as:

$$s_{\text{POS}_i} = \left\lceil \frac{n_i}{5} \right\rceil$$

where n_i is the number of lexical units with POS equal to POS_i in the given frame. For example, in the frame ARRIVING, if there are $n_{\text{verb}} = 15$ verb lexical units:

$$s_{\text{verb}} = \left\lceil \frac{15}{5} \right\rceil = 3$$

then the total number of samples for that POS is:

$$s_{\text{total}} = s_{\text{verb}} \times k = 3 \times 9 = 27$$

Similarly, for $n_{\text{noun}} = 8$, we compute:

$$s_{\text{noun}} = \left\lceil \frac{8}{5} \right\rceil = 2 \quad \Rightarrow \quad s_{\text{noun, total}} = 2 \times 9 = 18$$

This sampling strategy ensures a consistent balance of examples across frames and tasks, while maintaining semantic relevance and avoiding annotation redundancy.

D Linguistic Variations

A key aspect of our data construction process is the use of diverse linguistic templates for both questions and answers. Each template contains placeholders—such as X for the frame name or def(X) for the frame definition—that are instantiated using FrameNet annotations during prompt generation. This approach promotes generalization, prevents the model from memorizing fixed surface forms, and closely mirrors the variability found in real-world user queries.

Tables 4 and 5 provide concrete examples of linguistic variation for a representative open-ended task: frame definition. For each data split (training, validation, test), we sample distinct phrasings, ensuring that the same question or answer formulation is never shared across different splits. This careful partitioning avoids data leakage and tests the model's ability to generalize across different linguistic realizations.

Training
Q1: What is the definition of the X frame?
Q2: Can you define the X frame?
Q3: How is the X frame defined?
Validation
Q4: Could you provide the definition of the X frame?
Test
Q5: Please can you provide the definition of the X frame?

Table 4: Examples of question template variations for the frame definition task, grouped by data split. Each formulation is unique to a split to ensure maximal linguistic diversity and strict separation between training, validation, and test sets.

Training
Ans1: We can define the X frame as "def(X)."
Ans2: The X frame can be defined as "def(X)."
Ans3: The definition of the X frame is "def(X)."
Ans4: If we define the X frame, it would be "def(X)."
Ans5: We can define the X frame as "def(X)."
Ans6: The X frame can be defined as follows: "def(X)."
Ans7: We can outline the definition of the X frame as "def(X)."
Ans8: The X frame is defined as "def(X)."
Ans9: The X frame can be described as "def(X)."
Ans10: The definition of the X frame is "def(X)."
Validation
Ans11: The X frame can be summarized as "def(X)."
Ans12: If we describe the X frame, it will be "def(X)."
Ans13: A possible definition of the X frame is "def(X)."
Test
Ans14: The definition of the X frame can be "def(X)."
Ans15: We could define the X frame as "def(X)."

Table 5: Examples of answer template variations for the frame definition task, grouped by data split. As with the questions, each answer formulation is assigned to a single split, ensuring the model cannot rely on surface-level memorization.

E Task-level Evaluation Breakdown

To better understand where our frame-aware supervision strategy yields the most impact, we provide a task-level breakdown of performance. Table 7 reports scores on each of the 11 prompt-based tasks, comparing the original LLaMA model (zero-shot) with its fine-tuned counterpart. Task types are categorized as Open-ended, Closed-ended, or Multiple-choice (MCQs), and evaluated using appropriate metrics: cosine similarity for generative outputs, and F1 score for classification tasks.

The results consistently confirm the effectiveness of semantic supervision: for each task, the fine-tuned model outperforms the baseline. Particularly notable gains are observed in closed tasks requiring precise frame-role or definition-role verification (e.g., T3, T7, T9, T11), and in open-ended

ID	Prompt type	What varies?	Instances per frame	Why that number of instances?
T1	Open-ended	Wording of question about frame definition	5	One definition, asked in 5 paraphrased forms
T2	Open-ended	Wording of question about frame name from definition	5	One definition, reversed as 5 distinct questions
T3	Closed-ended	Match/mismatch of frame definitions	9 (3+6)	3 correct, 6 distractors from unrelated frames
T4	Open-ended	Surface forms of request for FE list	5	FE list is fixed, asked in 5 paraphrased forms
T5	Open-ended	Subset of core FEs and question formulation	5	2 FEs randomly sampled; question paraphrased 5 times
T6	Multiple-choice	Set of distractors for correct FE	5	Each with 1 correct + 3–4 distractors; mix of correct/incorrect MCQs
T7	Closed-ended	Pairings of roles with frames	9 (3+6)	3 true role sets, 6 sampled from unrelated frames
T8	Open-ended	Question formulation per core FE	$3 \times \lvert \text{core FE} \rvert$	3 paraphrases per core FE definition
T9	Closed-ended	FE–definition pairs	$9 \times \lvert \text{core FE} \rvert$	3 correct, 3 incorrect per FE
T10	Open-ended	POS-based sublists of LUs	$\sum_p \lceil \frac{n_p}{5} \rceil$	One question per LU-POS bucket; no overlap across splits
T11	Closed-ended	LU–POS verification questions	$9 \times \lceil \frac{n_p}{5} \rceil$	3 positives, 6 negatives per LU-POS bucket

Table 6: Summary of task-specific generation strategies and sample counts per frame. Task IDs correspond to the columns of Table 9.

ID	Task	Type	Metrics	Os	FT
T1	What is the definition of the X frame?	Open-ended	Cos	0.70	0.96
T2	Which frame is defined by def(X)?	Open-ended	Cos	0.50	0.84
T3	Is "def(X)" regarded as the definition of Frame X?	Closed	F1	0.48	0.70
T4	Can you list some frame elements in the X frame?	Open-ended	Cos	0.58	0.88
T5	Which is the frame involving frame elements such as FE$_1$ and FE$_2$?	Open-ended	Cos	0.47	0.83
T6	Which one of the following roles belongs to the set of frame elements of the X frame?	MCQs	F1	0.27	0.66
T7	Are roles such as Y and Z part of the frame elements of Frame X?	Closed	F1	0.61	0.85
T8	How is frame element FE$_i$ defined in X frame?	Open-ended	Cos	0.68	0.87
T9	Does the definition of frame element "def(FE$_i$)" accurately express FE$_i$ in the X frame?	Closed	F1	0.51	0.99
T10	Could you list some lexical units associated with the X frame?	Open-ended	Cos	0.75	0.85
T11	Can LU$_i$, as a POS$_i$, be considered as the lexical unit of the X frame?	Closed	F1	0.60	0.98

Table 7: Task-wise evaluation results across the 11 prompt templates. Metrics are cosine similarity for open-ended prompts and F1 score for closed-ended and MCQ formats.

prompts involving structured natural language responses (e.g., T1, T5, T8).

F Instruction-style Prompt Used for SRL Evaluation

We evaluate our model's semantic role labeling (SRL) capabilities using a controlled instruction-style prompt, shown in Figure 2. The prompt requires the model to extract both the lexical unit evoking the frame and the associated frame elements, returning a structured `JSON` object. The input consists of a sentence and its corresponding frame label.

Although the prompt includes a single illustrative example, this is not intended as one-shot learning: the example solely clarifies the expected output format and does not correspond to the frame used in the actual input. A true one-shot setting would require frame-specific exemplars for each evaluation case, which are not provided. Thus, the evaluation remains fully zero-shot with respect to frame-specific role assignments.

```
"""
You are an expert in Frame Semantics and Semantic Role Labeling. Your task is to identify the **
     lexical unit** evoking a given frame  and extract the corresponding **frame elements** with
     their roles from a given sentence.

### Instructions:
1. Identify the **lexical unit** that evokes the given frame.
2. Extract **frame elements** present in the sentence and map them to their respective roles
3. Format your response strictly as a JSON object following the structure provided.
4. Do not include any additional explanations-return only the JSON.
5. Use only **frame elements** you know.

### Example:
#### Given Frame: LOCATION
#### Input Sentence:
Hall, who recently returned from a trip to Iraq....
#### Expected Output:
{
    "input_sentence": "Hall, who recently returned from....",
    "annotations": [
        {
            "frame": "LOCATION",
            "lexical_unit": "trip",
            "frame_elements": {
                "PLACE": "Iraq",
                "TRAVELER": "Hall"
            }
        }
    ]
}
Now, process the following input and return a JSON object:

#### Given Frame:
#### Input Sentence:
#### Your Output:
"""
```

Figure 2: Instructional prompt used for SRL evaluation.

Frame Name	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	**Total**
RELEASING	5	5	9	5	5	5	9	9	27	2	18	99
MANIPULATION	5	5	9	5	5	5	9	9	27	2	18	99
CONTROL	5	5	9	5	5	5	9	21	63	2	18	147
KIDNAPPING	5	5	9	5	5	5	9	6	18	3	27	97
COMMUNICATION MANNER	5	5	9	5	5	5	9	9	27	2	18	99
CORPORAL PUNISHMENT	5	5	9	5	5	5	9	9	27	2	18	99
SUICIDE ATTACK	5	5	9	5	5	5	9	6	18	1	9	77
MOTION NOISE	5	5	9	5	5	5	9	15	45	1	9	113
COMMERCE PAY	5	5	9	5	5	5	9	15	45	2	18	123
RECEIVING	5	5	9	5	5	5	9	9	27	2	18	99
Total	50	50	90	50	50	50	90	108	324	19	171	1052

Table 8: Summary of the sample counts produced for each task across the unseen frames.

Frame Name	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	**Total**
Abandonment	5	5	9	5	5	5	9	6	18	12	27	106
Abusing	5	5	9	5	5	5	9	6	18	12	27	106
Appointing	5	5	9	5	5	5	9	15	45	9	27	139
Arrest	5	5	9	5	5	5	9	12	36	9	27	127
Arriving	5	5	9	5	5	5	9	6	18	11	45	123
Assemble	5	5	9	5	5	5	9	12	36	4	9	104
Assistance	5	5	9	5	5	5	9	12	36	13	36	140
Attack	5	5	9	5	5	5	9	6	18	18	81	166
Attempt means	5	5	9	5	5	5	9	9	27	4	9	92
Bringing	5	5	9	5	5	5	9	21	63	17	72	216
Come together	5	5	9	5	5	5	9	12	36	6	27	124
Committing crime	5	5	9	5	5	5	9	6	18	8	18	93
Communication	5	5	9	5	5	5	9	12	36	10	36	137
Communication response	5	5	9	5	5	5	9	15	45	10	36	149
Contacting	5	5	9	5	5	5	9	15	45	12	54	169
Departing	5	5	9	5	5	5	9	6	18	9	27	103
Enforcing	5	5	9	5	5	5	9	9	27	8	18	105
Escaping	5	5	9	5	5	5	9	6	18	9	27	103
Event	5	5	9	5	5	5	9	9	27	8	18	105
Eventive affecting	5	5	9	5	5	5	9	6	18	5	18	90
Examination	5	5	9	5	5	5	9	15	45	8	18	129
Exchange	5	5	9	5	5	5	9	6	18	8	18	93
Execute plan	5	5	9	5	5	5	9	9	27	8	18	105
Execution	5	5	9	5	5	5	9	6	18	9	27	103
Funding	5	5	9	5	5	5	9	9	27	4	9	92
Gesture	5	5	9	5	5	5	9	15	45	9	27	139
Getting	5	5	9	5	5	5	9	6	18	10	36	113
Giving	5	5	9	5	5	5	9	9	27	12	54	145
Intentionally act	5	5	9	5	5	5	9	6	18	10	36	113
Intentionally affect	5	5	9	5	5	5	9	9	27	4	9	92
Killing	5	5	9	5	5	5	9	15	45	24	135	262
Motion	5	5	9	5	5	5	9	21	63	8	45	180
Objective influence	5	5	9	5	5	5	9	21	63	8	18	153
Piracy	5	5	9	5	5	5	9	9	27	12	27	118
Rape	5	5	9	5	5	5	9	9	27	12	27	118
Replacing	5	5	9	5	5	5	9	9	27	9	27	115
Residence	5	5	9	5	5	5	9	9	27	15	54	148
Response	5	5	9	5	5	5	9	12	36	8	18	117
Rewards and punishments	5	5	9	5	5	5	9	12	36	12	27	130
Self motion	5	5	9	5	5	5	9	18	54	45	297	457
Smuggling	5	5	9	5	5	5	9	15	45	12	27	142
Summarizing	5	5	9	5	5	5	9	9	27	8	18	105
Supply	5	5	9	5	5	5	9	12	36	9	27	127
Supporting	5	5	9	5	5	5	9	6	18	4	9	80
Taking	5	5	9	5	5	5	9	9	27	8	18	105
Temporary stay	5	5	9	5	5	5	9	12	36	9	27	127
Theft	5	5	9	5	5	5	9	12	36	20	99	210
Using	5	5	9	5	5	5	9	12	36	13	36	140
Vehicle landing	5	5	9	5	5	5	9	6	18	4	9	80
Visiting	5	5	9	5	5	5	9	6	18	8	18	93
Total	250	250	450	250	250	250	450	525	1,575	524	1,854	6,628

Table 9: Overview of the number of samples generated for each task across frames, with tasks represented as columns and frames as rows. The table also includes the total number of sample pairs for each frame. Each cell reflects the actual number of QA pairs generated, which may vary according to (a) the number of frame elements or lexical units per frame, (b) the mix of positive and negative samples, and (c) the paraphrasing strategy adopted for data splitting. These design choices are fully detailed in Appendix B (for open-ended tasks) and Appendix C (for closed-ended tasks)

From Complex Word Identification to Substitution: Instruction-Tuned Language Models for Lexical Simplification

Tonghui Han[1,3], **Xinru Zhang**[2*], **Yaxin Bi**[1], **Maurice Mulvenna**[1] **and Dongqiang Yang**[2†]
[1] Ulster University, Belfast BT15 1ED, UK
[2] Shandong Jianzhu University, Jinan 250101, China
[3] Binzhou Polytechnic, Binzhou 256603, China
`{han-t, y.bi, md.mulvenna}@ulster.ac.uk`
`xinru.zhang198@gmail.com` `ydq@sdjzu.edu.cn`

Abstract

Lexical-level sentence simplification is essential for improving text accessibility, yet traditional methods often struggle to dynamically identify complex terms and generate contextually appropriate substitutions, resulting in limited generalization. While prompt-based approaches with large language models (LLMs) have shown strong performance and adaptability, they often lack interpretability and are prone to hallucinating. This study proposes a fine-tuning approach for mid-sized LLMs to emulate the lexical simplification pipeline. We transform complex word identification datasets into an instruction–response format to support instruction tuning. Experimental results show that our method substantially enhances complex word identification accuracy with reduced hallucinations while achieving competitive performance on lexical simplification benchmarks. Furthermore, we find that integrating fine-tuning with prompt engineering reduces dependency on manual prompt optimization, leading to a more efficient simplification framework.

1 Introduction

Lexical simplification (LS) aims to enhance text readability and understandability by replacing complex words or phrases with simpler alternatives without shifting the original meaning or introducing grammatical errors. Traditional LS methodologies typically follow the pipeline, including identifying complex components, finding substitution candidates, and choosing the optimal candidate (Qiang et al., 2020; Lee and Yeung, 2019; Atharva et al., 2023; Paetzold and Specia, 2017; Glavaš and Štajner, 2015). Although this strategy is widely adopted, it suffers from fundamental limitations in both design and execution. In particular, the sequential architecture is prone to error propagation, where errors in early steps negatively impact the quality of the subsequent outcomes. Consequently, accurately identifying complex words and phrases within a given sentence plays a pivotal role in the entire LS system. However, existing LS systems identify complex words by evaluating each word separately. The word semantic complexity relies on its context in real-world linguistic environments. The isolated evaluation strategy tends to reduce the recall of the complex word identification (CWI) module, lowering overall system performance. Although n-gram-based tokenization strategies can partially alleviate this issue, this static segmentation scheme limits its effectiveness in dealing with dynamic language environments (Ferrés et al., 2017).

Furthermore, previous approaches face challenges in generating appropriate substitutions and effectively ranking them. Static approaches, such as vector space-based approaches (Glavaš and Štajner, 2015; Ferrés et al., 2017), rank the substitute candidates by measuring the lexical semantic distance, which may result in semantic drift (Paetzold and Specia, 2017) or part-of-speech (PoS) tag mismatching (Glavaš and Štajner, 2015). Although dynamic approaches mitigate these limitations by leveraging pre-trained language models that utilize contextual knowledge to generate substitutes (Qiang et al., 2020; Atharva et al., 2023), similar to static approaches, these models are constrained to replace complex terms using isolated lexical units, which results in limited effectiveness in handling complex multi-word terms. As shown later in Section 4.2, our fine-tuned system is competitive with traditional models in preserving both grammaticality and adequacy.

Large language models have demonstrated remarkable language understanding capabilities in text generation tasks. Using LLMs enables effective adaptation to diverse linguistic contexts, efficiently identifying complex words and phrases, and generating appropriate substitutions by analyzing

---

[*] Equal contribution.

[†] Corresponding author.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 48–58
November 8-9, 2025 ©2025 Association for Computational Linguistics

the entire context (Baez and Saggion, 2023; North et al., 2024). However, since LLMs suffer from hallucinations, inconsistency, and limited controllability, depending on prompting alone is unreliable for LS tasks. Moreover, LLMs are sensitive to the prompt format, leading to an extensive manual design and tuning, making the development process both time-consuming and resource-intensive.

This paper proposes an LS approach that fine-tunes LLMs on a modified corpus and instructs the model to identify and simplify complex terms within given sentences. The experimental results demonstrate that the proposed system can identify variable-length phrases or single words by leveraging its advanced language understanding capability. It simultaneously replaces all complex terms during simplification, while avoiding semantic drift arising from iterative substitution processes in traditional approaches. Moreover, this system addresses the limitations of purely prompt-based methods. Since our system only requires the user to write the instructions in the same format as the training corpora, it does not require users to engage in tuning prompt format. Besides, the fine-tuning process effectively enhances the accuracy rate with decreasing hallucinations. To enable LLMs to comprehend and execute instructions precisely, we construct an LS-instruction-answer (LS-I-A) dataset based on the CWIG3G2 English dataset (Yimam et al., 2017). Experimental results indicate that the proposed approach, integrating fine-tuning with a prompt mechanism, achieves superior simplification efficiency compared with traditional LS pipelines. Moreover, unlike few-shot and zero-shot methods, this approach eliminates the requirements for prompt tuning while effectively mitigating hallucination rates. The key contributions of this paper are as follows.

- We propose a two-step instruction-tuned framework for lexical-level sentence simplification, which explicitly separates complex word identification and lexical substitution, enabling more accurate, context-aware simplifications.

- We construct an instruction-based dataset by reformatting the CWIG3G2 (Yimam et al., 2017) corpus into structured CWI and LS instances, facilitating instruction tuning and downstream evaluation on lexical simplification.

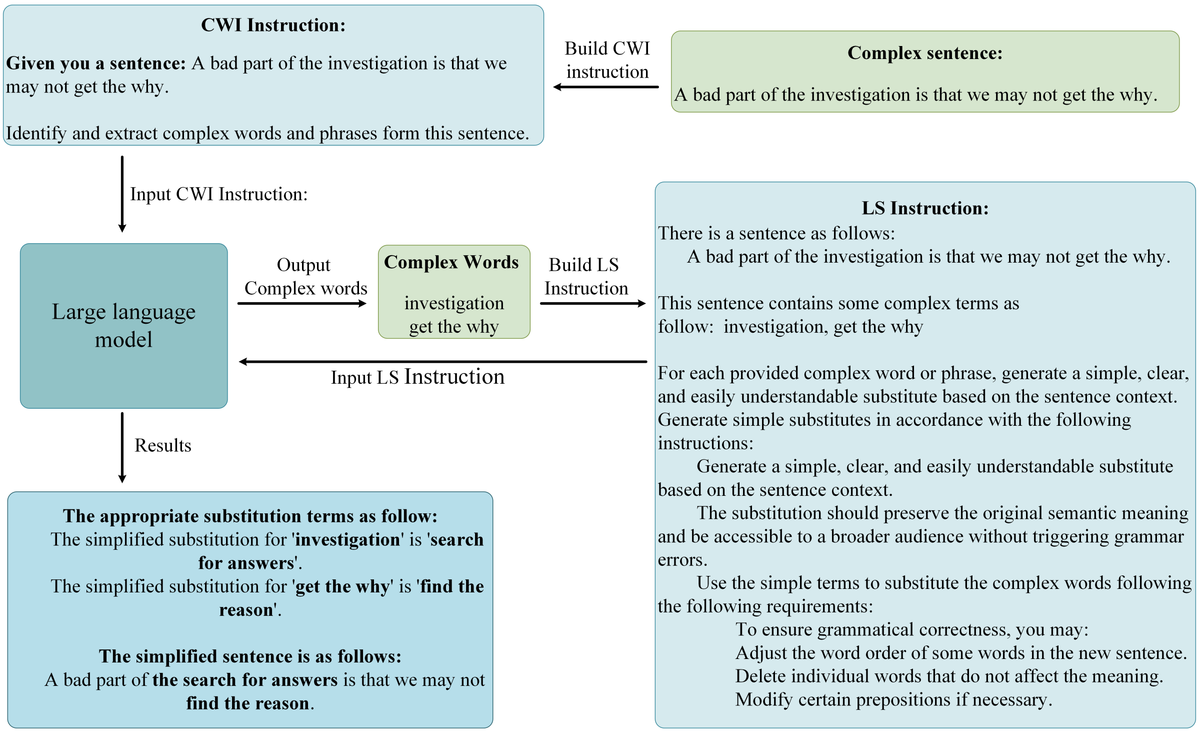
- We conduct comprehensive experiments across multiple LLM backbones (Mistral (Jiang et al., 2023), Qwen (Yang et al., 2025), and LLaMA (Touvron et al., 2023)), comparing fine-tuned and non-fine-tuned settings using both standard automatic metrics (Section 4).

2 Related Works

Traditional LS approaches follow a three-step pipeline—complex word identification, substitution generation, and ranking—yet suffer from error propagation and context-insensitive evaluations (Paetzold and Specia, 2017; Glavaš and Štajner, 2015). To address these issues, edit-based models such as EditNTS (Dong et al., 2019) and Edit-TS (Kumar et al., 2020) perform explicit token-level operations (e.g., delete, replace, keep), achieving practical simplification with interpretable edits. Similarly, GRS (Dehghan et al., 2022) combines generation and revision in an unsupervised manner, incorporating paraphrasing and deletion at the lexical level. Dress-LS (Zhang and Lapata, 2017) introduces a sequence-to-sequence model trained with reinforcement learning to optimize simplification quality, but offers limited control over specific linguistic properties. ACCESS (Martin et al., 2019) extends this by enabling controllable simplification through attribute-specific constraints such as length and lexical complexity. Despite these advances, many of these systems rely on rigid heuristics or lack the scalability to handle diverse inputs dynamically.

Recent work also explores the use of LLMs such as LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023) for lexical simplification (Baez and Saggion, 2023; North et al., 2024). With strong instruction-following and contextual understanding abilities, these models can simplify complex terms in context via prompting or fine-tuning, offering greater flexibility and scalability. However, they often suffer from limited interpretability and occasional hallucinations, making them less reliable for controlled LS tasks.

In contrast, our work leverages LLMs fine-tuned on a structured instruction-based dataset to identify and directly simplify complex terms in context. Unlike edit-based systems or prompt-only LLM approaches, our method balances controllability and robustness while mitigating hallucinations and format sensitivity.



Given you a sentence: A bad part of the investigation is that we may not get the why.

Identify and extract complex words and phrases form this sentence.

A bad part of the investigation is that we may not get the why.

There is a sentence as follows:
A bad part of the investigation is that we may not get the why.

This sentence contains some complex terms as follow: investigation, get the why

For each provided complex word or phrase, generate a simple, clear, and easily understandable substitute based on the sentence context. Generate simple substitutes in accordance with the following instructions:
Generate a simple, clear, and easily understandable substitute based on the sentence context.
The substitution should preserve the original semantic meaning and be accessible to a broader audience without triggering grammar errors.
Use the simple terms to substitute the complex words following the following requirements:
To ensure grammatical correctness, you may:
Adjust the word order of some words in the new sentence.
Delete individual words that do not affect the meaning.
Modify certain prepositions if necessary.

The appropriate substitution terms as follow:
The simplified substitution for '**investigation**' is '**search for answers**'.
The simplified substitution for '**get the why**' is '**find the reason**'.

The simplified sentence is as follows:
A bad part of **the search for answers** is that we may not **find the reason**.

Figure 1: The workflow of the LS system: it first identifies complex words using CWI instructions. Then, it builds LS instructions by utilizing the extracted complex words. Finally, the system outputs the simplified versions under the guidance of the constructed LS instructions.

3 Methodologies

We design a two-stage workflow for lexical simplification using a large language model, as Figure 1 shows. The first stage identifies complex words and phrases within a given sentence. In the second stage, the LLM is prompted with the original sentence and the list of identified complex terms. It is instructed to generate simpler, contextually appropriate alternatives that preserve the original meaning and ensure grammatical correctness. The model may also perform minor syntactic adjustments such as reordering words, modifying prepositions, or removing redundant tokens to maintain fluency. This approach leverages the LLM's contextual understanding and generation capabilities to perform high-quality, semantic-preserving simplification without requiring manually crafted rules or external linguistic resources.

In the following sections, we detail methodologies used in designing this LS system, including constructing the LS-I-A corpus, fine-tuning strategy for LLMs, and evaluation metrics.

3.1 LS-I-A Dataset

3.1.1 Complex Word Identification

We build on the CWI instruction dataset based on the CWIG3G2 English dataset (Yimam et al., 2017) which contains human annotations of complex words across various genres such as News, WikiNews, and Wikipedia. In this study, we manually alter the dataset to transfer the original dataset into instruction-answer format. To convert the dataset into an instruction–answer format suitable for prompting LLMs, we manually revise each instance. This process involves the following steps:

- Sentence reformatting: we extract sentences from the original tabular format and integrate them into instructional prompts as is presented in Figure 1 (CWI Instruction).
- Annotation consolidation: All annotated complex terms within each sentence are aggregated and listed in the answer as follows.

  `The complex words and phrases: investigation, get the why.`

3.1.2 Lexical Simplification

We further adapt the dataset for lexical simplification. While CWIG3G2 was originally designed for CWI, we extend it via GPT-assisted simplification and expert validation, converting it into an instruction-tuning dataset for LS. First, we employ GPT-4o to generate appropriate simple substitute candidates for each complex term. Next, the complex terms are replaced with their corresponding substitutes. To ensure adequacy and grammatical correctness, we engage native English speakers to assess and correct the simplified sentences. Simultaneously, non-native English speakers with experience in English language teaching in China assess the readability of the simplified terms for non-native readers. Following the evaluation and revision processes, we retain 1,784 instances in the training set and 225 instances in the validation set. The final dataset is formatted into an instruction-answer format with the following structure.

- Instruction: It contains the original complex sentence, followed by identified complex terms. The instruction also specifies that each complex term should be simplified based on contextual knowledge. Furthermore, we outline certain restrictions to keep the original meaning and avoid grammatical errors. The instance is detailed in Figure 1 (LS Instruction)

- Answer: The answer should list simple substitution terms and present the simplified sentence.

Therefore, the LS-I-A dataset comprises four groups of attributes. The attributes `cwi_instruction` and `cwi_answer` provide instructions and references designed for fine-tuning LLMs to identify complex terms. Similarly, `ls_instruction` and `ls_answer` are constructed to guide the fine-tuning process for lexical simplification.

3.2 Fine-tuning

Notably, our fine-tuning is performed on a small-scale dataset, demonstrating the feasibility of low-resource instruction tuning for lexical simplification. In the training process, to enable the model to simulate the LS pipeline, for each instance, its `cwi_instruction` is first fed into the LLMs, immediately followed by its `ls_instruction`. All the instances are iteratively fed into the LLMs following these steps, enabling them to learn the simplification rules and strategies.

We use our newly constructed dataset to fine-tune LLMs for complex word identification and lexical simplification. In this study, we select three mid-sized open-source instruction-aligned LLMs: Llama-3.1-8B-Instruct[1], Qwen2.5-7B-Instruct-1M[2], and Mistral-8B-Instruct-2410[3].

All models are fine-tuned using **LoRA** (Low-Rank Adaptation) with 4-bit precision (NF4 quantization) to enhance memory efficiency. The fine-tuning process is executed on an **RTX 4090 GPU**, with gradient checkpointing enabled to reduce memory consumption. The models are trained for 10 epochs with a per-device batch size of 2 and a gradient accumulation of 4. To optimize performance, a cosine learning rate scheduler with an initial learning rate of 1e-4 is applied, complemented by mixed precision (fp16). The optimizer is configured as **AdamW**, with a weight decay of 0.05 and a maximum gradient norm of 0.3. Model checkpoints and evaluations are performed at the end of each epoch, and the best-performing model is restored after the completion of training.

3.3 Evaluation Metrics

For CWI tasks, information retrieval metrics (precision, recall, and F1-score) are employed to evaluate the systems, with the basic metrics illustrated by Manning (2009). In addition to these basic information retrieval metrics, we introduce a novel evaluation metric, hallucination rate (HR), to assess the reliability of the LLMs. To obtain the HR, we define a hallucination term as follows.

A **hallucination term** is defined as any term generated by the LLMs that either does not appear in the original sentence or is identified as a numerical value, special character, or stop word.

The equation for calculating HR is presented below.

$$HR = \frac{1}{n}\left[\sum_{i=1}^{n}\frac{h_i}{m_i}\right] \quad (1)$$

where, $\boldsymbol{n}$ represents the test size, h_i denotes the number of hallucination terms in instance $\boldsymbol{i}$, and m_i is the number of identified terms in instance $\boldsymbol{i}$.

We evaluate the effectiveness of our system on the lexical-level sentence simplification task using three established metrics: SARI (Xu et al., 2016) for simplicity, FKGL (Flesch–Kincaid Grade

---

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M

[3] https://huggingface.co/mistralai/Ministral-8B-Instruct-2410

Level) (Kincaid et al., 1975) for fluency, and $\text{BERT}_{\text{score}}$ (Zhang et al., 2019) for adequacy. SARI and FKGL are obtained through the EASSE evaluation toolkit (Stodden, 2024), which provides a standardized framework for assessing simplification quality, while $\text{BERT}_{\text{score}}$ (Zhang et al., 2019) is calculated via the bert-score module[4].

4 Results

4.1 CWI results

The reformulated CWI task requires the model to extract all complex terms in a sentence rather than focusing on preselected target words, thereby expanding the instance space and altering the task definition. Consequently, direct comparison with traditional CWI methods is not methodologically valid. Instead, we investigate the effectiveness of instruction-based fine-tuning by contrasting it with non-fine-tuned counterparts under identical conditions.

We revise the CWIG3G2 test set following the methodology outlined in Section 3.1.1 to assess the performance of our system on CWI tasks. The resulting test set comprises 325 instances, each consisting of an instruction paired with a corresponding reference. The evaluation results are presented in Table 1. As demonstrated, the fine-tuned LLaMA model achieves the highest F1-score (0.8188), along with notable precision (0.7784) and the highest recall (0.8635), indicating a balanced and highly effective identification of complex words. The fine-tuned Mistral model also performs strongly, with an F1-score of 0.8064 and the lowest hallucination rate of 0.0055, demonstrating its reliability. Additionally, the fine-tuned Qwen model shows competitive performance, achieving an F1-score of 0.8079 and the highest precision (0.7864) among all fine-tuned models while maintaining a reasonable recall (0.8306).

In contrast, the non-fine-tuned versions of all three models yield substantially lower F1 scores and significantly higher hallucination rates. For example, non-fine-tuned Qwen, despite achieving the highest precision overall (0.8388), suffers from a low recall (0.4727), resulting in a significantly lower F1-score (0.6077). These results underscore the effectiveness of fine-tuning in enhancing the accuracy and robustness of LLMs for CWI tasks.

4.2 Simplification Results

This section presents the simplification results of our system and provides a comparative analysis against state-of-the-art lexical simplification models as well as non-fine-tuned LLMs. All the LS systems are evaluated on the TurkCorpus (Xu et al., 2016) and ASSET (Alva-Manchego et al., 2020) datasets. This study selects NTS+SARI (Nisioi et al., 2017), ACCESS (Martin et al., 2019), EditNTS (Dong et al., 2019), Edit-Unsup-TS (Kumar et al., 2020), Dress-LS (Zhang and Lapata, 2017), GRS (Dehghan et al., 2022), non-fine-tuned LLaMA, non-fine-tuned Qwen, and non-fine-tuned Mistral as baseline models due to their relevance to lexical-level sentence simplification. The evaluation results are summarized in Table 2 for clarity and comparison.

As shown in Table 2, our fine-tuned models demonstrate competitive or superior performance compared with both traditional lexical simplification systems and non-fine-tuned LLMs. Notably, the fine-tuned Qwen model achieves the highest SARI score on the ASSET benchmark (41.41), along with a $\text{BERT}_{\text{score}}$ of 0.9492 and an FKGL of 7.568, indicating its strong capability in producing simplified text while preserving semantic content. Fine-tuned Mistral also performs well, with SARI scores of 39.55 on TurkCorpus and 41.24 on ASSET, $\text{BERT}_{\text{score}}$s of 0.942 and 0.953, and an FKGL of 7.32 on both datasets. Similarly, fine-tuned LLaMA attains SARI scores of 39.13 on TurkCorpus and 40.59 on ASSET, with corresponding $\text{BERT}_{\text{score}}$s of 0.941 and 0.950.

Among traditional systems, ACCESS achieves the best performance on TurkCorpus, with the highest SARI score (42.08) and the lowest FKGL (7.29). However, its $\text{BERT}_{\text{score}}$ (0.955) remains slightly lower than those of EditNTS (0.961) and Dress-LS (0.964). On ASSET, GRS yields the lowest FKGL (4.17), indicating enhanced readability, while EditNTS achieves the highest $\text{BERT}_{\text{score}}$ (0.970), reflecting strong semantic fidelity.

Overall, these results confirm the effectiveness of fine-tuning large language models for lexical simplification. All fine-tuned models consistently outperform their non-fine-tuned counterparts and match or exceed the performance of established simplification systems.

---

[4] https://github.com/Tiiiger/bert_score

model	precision ↑	recall ↑	F1 ↑	HR ↓
ft Mistral	0.7667	0.8545	0.8064	**0.0055**
ft Qwen	0.7864	0.8306	0.8079	0.0145
ft LLaMA	0.7784	**0.8635**	**0.8188**	0.0065
non-ft Mistral	0.6207	0.7388	0.6778	0.0616
non-ft Qwen	**0.8388**	0.4727	0.6077	0.0335
non-ft LLaMA	0.7300	0.6712	0.6994	0.0532

Table 1: Evaluation results for the CWI task, comparing fine-tuned models with their original versions. Background color indicates performance level (darker green = better; lower is better for HR).

Model	**TurkCorpus**			**ASSET**		
	SARI ↑	**FKGL ↓**	**BERTscore ↑**	**SARI ↑**	**FKGL ↓**	**BERTscore ↑**
NTS+SARI (Nisioi et al., 2017)	36.93	8.18	0.959	34.02	8.18	0.967
ACCESS (Martin et al., 2019)	**42.08**	7.29	0.955	40.12	7.29	0.966
EditNTS (Dong et al., 2019)	38.51	8.37	0.961	34.94	8.37	**0.970**
Edit-Unsup-TS (Kumar et al., 2020)	38.09	6.44	–	38.94	6.39	–
Dress-LS (Zhang and Lapata, 2017)	36.89	7.58	**0.964**	36.90	7.58	0.951
GRS (Dehghan et al., 2022)	–	–	–	37.9	**4.17**	–
non-ft Mistral	33.43	8.99	0.924	38.69	8.99	0.936
non-ft Qwen	32.61	**5.24**	0.901	39.13	5.24	0.915
non-ft LLaMA	34.68	7.47	0.921	39.71	7.47	0.932
ft Mistral	39.55	7.32	0.942	41.24	7.32	0.953
ft Qwen	38.60	7.57	0.940	**41.41**	7.568	0.949
ft LLaMA	39.13	7.59	0.941	40.59	7.59	0.950

Table 2: Performance of different models on TurkCorpus and ASSET benchmarks. Cell background color indicates performance: darker green = better. For FKGL, lower values are better and mapped to deeper green.

5 Analysis

As shown in Table 1 and Table 2, the instruction-tuned models exhibit substantial improvements over their non-fine-tuned counterparts on the CWI and LS tasks. These results suggest that instruction-based fine-tuning enhances the ability of large language models to identify complex lexical items with greater accuracy. Furthermore, our system is competitive with existing systems, generating simplifications that are more fluent and contextually aligned. These findings underscore the effectiveness of our framework in steering LLMs toward more precise, context-aware, and semantically faithful simplification.

To gain deeper insight into the behavior of our instruction-tuned lexical simplification system, we analyze the SARI subcomponents alongside the structural and lexical metrics presented in Figures 2 and 3. These fine-grained evaluations go beyond aggregate performance scores, offering a more nuanced understanding of how the model balances adequacy, fluency, and simplicity in its simplification strategy.

5.1 SARI Subcomponent Analysis

As shown in Figure 2, our fine-tuned models exhibit competitive performance across the three SARI subcomponents—add, keep, and delete—demonstrating the effectiveness of our system in capturing the diverse operations involved in sentence simplification. Notably, the fine-tuned LLaMA model achieves the highest add score on TurkCorpus and maintains strong performance on ASSET, indicating that it is particularly effective at introducing simplified content that aligns well with the surrounding context. This capability is associated with our instruction tuning framework, which separates complex word identification from substitution generation. By decoupling these stages, the model is first guided to identify complex items by leveraging word morphology and contextual cues, and subsequently to generate replacements that are better informed by the identified term and its contextual knowledge, resulting in more appropriate

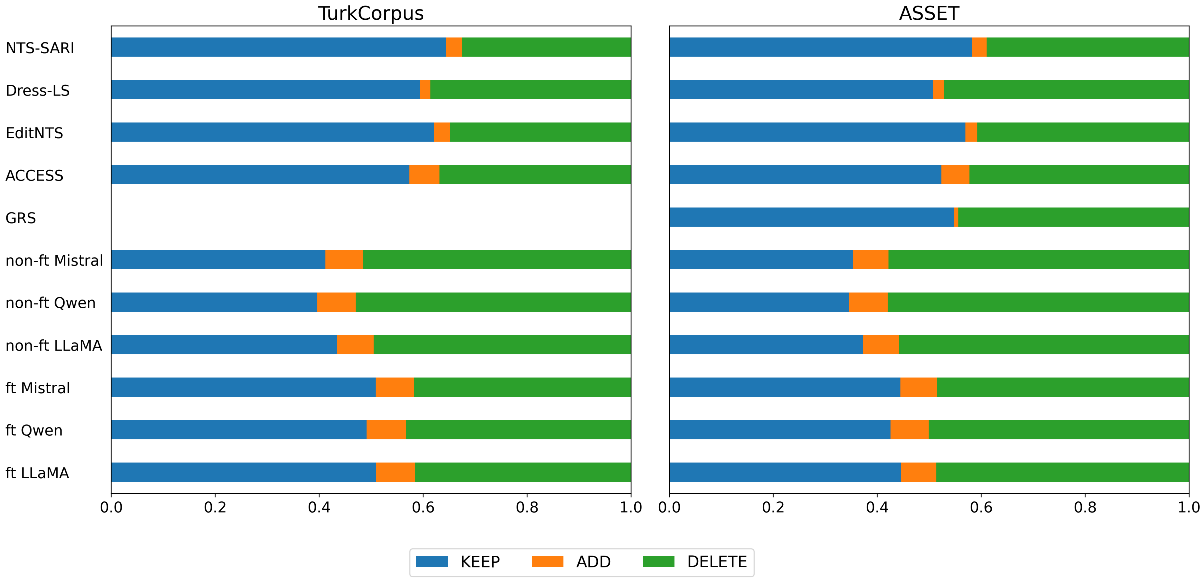


Figure 2: SARI subcomponent ratio (add, keep, delete) for TurkCorpus and ASSET datasets across different models.

and semantically coherent simplifications.

Although the keep scores of our fine-tuned models are lower than those of traditional systems such as EditNTS and ACCESS, they reflect a more transformative simplification strategy. While higher keep scores generally indicate stronger preservation of the original sentence structure, they may also reflect a more conservative editing approach with limited modification. In contrast, our models are more inclined to rephrase the complex items and partially restructure the input, yielding moderate keep scores that maintain core semantic content while enabling both lexical and syntactic simplification. This editing behavior is further supported by consistently strong delete scores, which suggest that our models are effective at removing unnecessary or overly complex content, thereby enhancing clarity and conciseness.

5.2 Structural and Lexical Evaluation

Figure 3 offers a visual demonstration of how instruction-based fine-tuning enhances the quality of text simplification along both structural and lexical dimensions. Compared to baseline models, our fine-tuned models exhibit a balanced distribution across key structural indicators. Notably, the compression ratio (blue segments) for the fine-tuned models remains close to 0.90, indicating that they reduce sentence length effectively without aggressive truncation. In contrast, non-fine-tuned models display more extreme behaviors—for example, non-fine-tuned Qwen heavily compresses (left-shifted blue segment), while non-fine-tuned Mistral barely shortens inputs at all.

Sentence split proportions (orange segments) are consistent across all models, with fine-tuned variants maintaining structural cohesion. The "Exact copies" segments (green) are minimal for fine-tuned models—remarkably fine-tuned LLaMA—highlighting their strong rewriting capability compared to traditional models like Dress-LS, which retain a higher proportion of copied content.

Additionally, the red (Additions) and cyan (Deletions) segments show that the fine-tuned models engage in more balanced and substantive edits. For example, fine-tuned LLaMA and Qwen demonstrate nearly symmetrical proportions of additions and deletions, indicating that the models are not merely replacing individual words (as in shallow lexical simplification) but are actively restructuring sentences by inserting relevant information and removing redundant or complex segments, thereby engaging in more meaningful and substantive simplification.

Finally, the Levenshtein similarity (gray segments) and lexical complexity scores indicate that our fine-tuned models strike a desirable balance between adequacy and fluency. The longer gray bars in fine-tuned models signify sufficient divergence from the source while maintaining coherence. These visual patterns affirm the benefits of fine-tuning LLMs in sentence simplification, aligning more closely with human-like simplification behaviors.

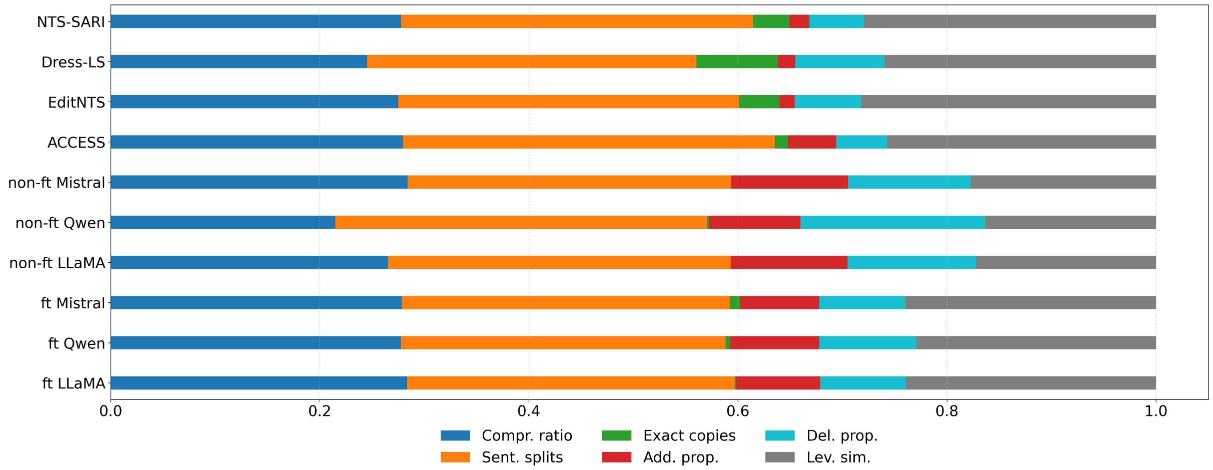


Figure 3: Distribution of Structural and Lexical Metrics Across Simplification Models.

In summary, these results validate the strength of our instruction-based fine-tuning approach. By structurally isolating CWI and LS within modular prompt designs, our models consistently outperform their non-fine-tuned counterparts and match or exceed the performance of existing state-of-the-art lexical simplification systems across various evaluation criteria, including simplification depth, fluency, and adequacy. Further illustrations are detailed in Appendix A, which includes three representative cases comparing traditional models, non-fine-tuned LLMs, and fine-tuned LLMs. The analysis demonstrates how instruction tuning enables more fluent, semantically faithful, and structurally appropriate simplifications across varied sentence types.

6 Conclusion

This study proposes a lexical-level sentence simplification system based on LLMs that emulate the traditional LS pipeline. We revise the CWIG3G2 dataset to develop this system and construct a new instruction–answer corpus, LS-I-A, for fine-tuning LLMs. Experimental results demonstrate that fine-tuning significantly enhances LLM performance in CWI and LS tasks. On LS tasks, our system achieves performance comparable to several state-of-the-art models. Despite the strong performance, our system still requires further improvements. Notably, while LS-I-A performs well in practice, it lacks reasons for explaining the selection of complex words and the generation of simplified alternatives. As a result, the system does not support Chain-of-Thought reasoning in either the identification or substitution stages, limiting the interpretability of its outputs. In future work, we aim to refine the LS-I-A dataset and incorporate strategies that enhance the system's reasoning capabilities.

Limitations

Our study has several limitations. First, the LS-I-A dataset is designed to fine-tune LLMs by mimicking the traditional LS pipeline, identifying complex words, and providing substitutions. However, it offers no explanations or rationales for simplifying choices, lacking interpretability. Moreover, the dataset does not consider stylistic factors; it primarily emphasizes semantic preservation and grammatical correctness, neglecting aspects such as tone or fluency. Second, although fine-tuning improves LLM performance on simplification tasks, the generation process remains largely uninterpretable and challenging to control. As a result, models can still produce hallucinations or overly aggressive simplifications that distort the original meaning. Finally, current evaluation metrics are insufficient for thoroughly assessing simplification quality. Sentence simplification involves multiple dimensions that are not comprehensively captured by existing metrics. In particular, SARI is heavily reference-dependent and can be biased by the lexical and stylistic preferences in the reference simplifications.

Acknowledgments

We gratefully acknowledge the support of Ulster University and Shandong Jianzhu University. Their generous assistance and institutional backing have been instrumental in enabling this research and contributing to its successful completion.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *arXiv preprint arXiv:2005.00481*.

Kumbhar Atharva, Sonawane Sheetal, Kadam Dipali, and Mulay Prathamesh. 2023. Casm-context and something more in lexical simplification. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 506–515.

Anthony Baez and Horacio Saggion. 2023. Lsllama: Fine-tuned llama for lexical simplification. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 102–108.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. Grs: Combining generation and revision in unsupervised sentence simplification. *arXiv preprint arXiv:2203.09742*.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. *arXiv preprint arXiv:1906.08104*.

Daniel Ferrés, Horacio Saggion, and Xavier Gómez Guinovart. 2017. An adaptable lexical simplification architecture for major ibero-romance languages. In *Proceedings of the first workshop on building linguistically generalizable NLP systems*, pages 40–47.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Yinhan Jiang and 1 others. 2023. Mistral 7b. `https://mistral.ai/news/announcing-mistral-7b/`.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. *arXiv preprint arXiv:2006.09639*.

John SY Lee and Chak Yan Yeung. 2019. Personalized substitution ranking for lexical simplification. In *Proceedings of the 12th international conference on natural language generation*, pages 258–267.

Christopher D Manning. 2009. *An introduction to information retrieval.*

Louis Martin, Benoît Sagot, Eric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: An end-to-end lexical simplification framework. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 1–11.

Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lsbert: A simple framework for lexical simplification. *arXiv preprint arXiv:2006.14939*.

Regina Stodden. 2024. Easse-de: Easier automatic sentence simplification evaluation for german. *arXiv preprint arXiv:2404.03563*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, and 1 others. 2025. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Cwig3g2-complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407.

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

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

Case 1	She was among the first doctors to object to cigarette smoking around children, and drug use in pregnant women.
NTS+SARI	She was among the first doctors to object to cigarette smoking around children, and drugs use in pregnant women.
ACCESS	She was one of the first doctors to find out how to cigarette smoking around children, and use in pregnant women.
EditNTS	She was among the first doctors to object to cigarette smoking around children, and drug use in pregnant women.
Dress-LS	She was among the first doctors to make cigarette smoking around children .
non-ft Mistral	She was a first medical professional to oppose smoking near kids and drug misuse during pregnancy.
non-ft Qwen	She was among the earliest doctors to oppose smoking near kids and taking drugs while pregnant.
non-ft LLaMA	She was among the first doctors who opposed cigarette smoking near children, and drug use when women were expecting a baby.
ft Mistral	She was among the first doctors to protest against smoking around children, and drug use in expecting women.
ft Qwen	She was among the first doctors to protest smoke around children, and drug use by expecting mothers.
ft LLaMA	She was among the first doctors to protest smoking around children, and drug use in expecting women.
Case 2	They are culturally akin to the coastal peoples of Papua New Guinea.
NTS+SARI	They are culturally akin to the coastal communities of Papua New Guinea.
ACCESS	They are to the coastal peoples of Papua New Guinea, Papua New Guinea.
EditNTS	They are akin to the coastal peoples of Papua New Guinea.
Dress-LS	They are culturally referring to the coastal peoples of Papua New Guinea.
non-ft Mistral	They have similar cultures to the people who live near the coasts of Papua New Guinea.
non-ft Qwen	They are similar to people who live near the coast in Papua New Guinea.
non-ft LLaMA	The coastal peoples of Papua New Guinea have much in common with them.
ft Mistral	They are similar in culture to the near-sea people of Papua New Guinea.
ft Qwen	They are similar to the people near the sea in Papua New Guinea.
ft LLaMA	They are similar in culture to the people near the sea in Papua New Guinea.
Case 3	These attacks may have been psychological in origin rather than physical.
NTS+SARI	These attacks may have been psychological in origin than physical.
ACCESS	These attacks may have been different in origin rather than physical.
EditNTS	These attacks may have been psychological in origin and physical.
Dress-LS	These attacks may have been psychological in origin rather than physical.
non-ft Mistral	These actions could have started mentally instead of physically.
non-ft Qwen	These assaults might have started in people's minds instead of causing direct harm.
non-ft LLaMA	These attacks may be caused by emotional pain instead of causing physical harm.
ft Mistral	These attacks may have been meant to scare people rather than hurt them.
ft Qwen	These attacks may have been mental in source rather than physical.
ft LLaMA	These attacks may have been mental cause rather than physical.

Table 3: Lexical simplification examples across three cases. Our instruction-tuned models produce simpler, more fluent outputs while retaining the original meaning.

A Appendix

Three representative examples are used to analyze the behavioral differences across simplification systems. The comparison includes outputs from traditional models, non-fine-tuned LLMs, and instruction-fine-tuned LLMs.

The first case contains multiple complex expressions joined by coordination. Traditional models such as Dress-LS and EditNTS perform primarily local edits, often deleting or replacing single-word units. This strategy frequently results in incomplete simplifications or loss of important content. Non-fine-tuned LLMs produce more syntactically varied outputs, but these are often semantically inaccurate. Instruction-fine-tuned models apply edits across multiple spans, preserving the core meaning while improving surface fluency. The instruction–answer format used during training presents simplification as a global transformation task, requiring the model to operate over the entire sentence rather than isolated tokens.

The second case includes a long noun phrase with generalized referents. Outputs from traditional models exhibit minimal restructuring and mostly retain the original phrasing. Non-fine-tuned LLMs attempt paraphrasing but often generate repetitive or verbose alternatives. Instruction-fine-tuned models produce simpler constructions that reduce lexical and syntactic complexity without distorting the original meaning. The presence of sentence-level, span-aligned annotations in the LS-I-A dataset provides direct supervision for such multi-span transformations, encouraging broader structural adjustment rather than surface-level replacement.

The third case features contrastive lexical elements and implicit logical relations. Traditional models often simplify only part of a contrastive expression or leave the contrast unclear, reducing semantic clarity in the output. Non-fine-tuned LLMs vary in output quality, often generating inconsistent or logically disjointed results. Instruction-fine-tuned models consistently preserve the contrast and simplify the associated expressions in a controlled manner. The simplification objective in training is framed around semantic preservation under minimal complexity, which supports stable handling of discourse-level relationships in cases involving contrast or attribution.

The observed differences across systems correspond to their supervision regimes. Traditional models rely on local alignment or rule-based editing, which constrains their capacity for structural rewriting. Non-fine-tuned LLMs lack explicit task grounding and produce unstable outputs. Instruction-fine-tuned LLMs receive training on task-specific instructions and span-level supervision, which enables more consistent simplification at both lexical and structural levels.

Semantic Prosody in Machine Translation: the English-Chinese Case of Passive Structures

Xinyue Ma[1,2], **Pol Pastells**[1,2], **Mireia Farrús**[1,2], **Mariona Taulé**[1,2]
[1]Centre de Llenguatge i Computacio (CLiC), Universitat de Barcelona, Spain
[2]Institut de Recerca en Sistemes Complesos (UBICS), Universitat de Barcelona, Spain
`{maxinyue, pol.pastells, mfarrus, mtaule}@ub.edu`

Abstract

Semantic prosody is a collocational meaning formed through the co-occurrence of a linguistic unit and a consistent series of collocates, which should be treated separately from semantic meaning. Since words that are literal translations of each other may have different semantic prosody, more attention should be paid to this linguistic property to generate accurate translations. However, current machine translation models cannot handle this problem. To bridge the gap, we propose an approach to teach machine translation models about semantic prosody of a specific structure. We focus on Chinese BEI passives and create a dataset of English-Chinese sentence pairs with the purpose of demonstrating the negative semantic prosody of BEI passives. Then we fine-tune OPUS-MT, NLLB-600M and mBART50 models with our dataset for the English-Chinese translation task. Our results show that fine-tuned MT models perform better on using BEI passives for translating unfavourable content and avoid using it for neutral and favourable content. Also, in NLLB-600M, which is a multilingual model, this knowledge of semantic prosody can be transferred from English-Chinese translation to other language pairs, such as Spanish-Chinese.

1 Introduction

The notion of semantic prosody was first proposed in corpus linguistics in the 1990s. It was Sinclair (1987) who first noticed the phenomenon that some words or phrases tend to co-occur with unpleasant events, such as *HAPPEN* and *SET IN*. This property was later named Semantic Prosody (an analogy to Firth (1964)’ s concept of phonological prosody) by Louw (1993) and defined as “a consistent aura of meaning with which a form is imbued by its collocates”. Another well-known example is *CAUSE* as a verb (Stubbs, 1995). As shown in Figure 1, entities “caused” are generally considered undesirable: rupture, death, pain, misery, damage, and diseases.

Sinclair (1996) considers semantic prosody to be on the “pragmatic side of the semantics/pragmatics continuum” and can reveal the speaker or writer’s attitude. However, semantic prosody cannot be precisely perceived by individual native speakers through intuition (Louw, 1993; McEnery et al., 2006). A statistical analysis of corpus data is needed to correctly interpret the semantic prosody of a linguistic unit (a node), since corpus data can provide multiple usage patterns across many speakers and reflect the general intuition of native speakers (Stubbs, 2001; McEnery et al., 2006; Stewart, 2010). The co-occurrence of a node and a predominantly positive/negative context can foster subtle associative meaning, and the node is then considered to have positive/negative semantic prosody.

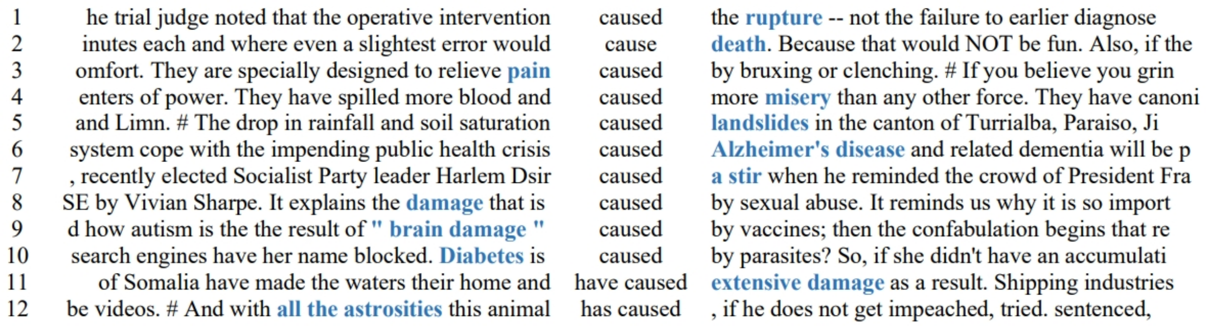
A word in English probably has a synonym in Chinese with the same semantic meaning. However, the two words do not necessarily have the same semantic prosody due to language variation. For example, the semantic meaning of the English word *BECAUSE* and the Chinese word “由于 *YOUYU*” are equivalent, but their semantic prosodies are not. *BECAUSE* has a neutral semantic prosody, while *YOUYU* has a negative one (Wu and Lan, 2019). Another example is *INSIST ON*, which is often used to describe annoying stubbornness, while its literal translation in Chinese “坚持 *JIANCHI*” has positive semantic prosody. This divergence in semantic prosody has even caused English learners who are native Mandarin Chinese speakers to use *INSIST ON* to mean encouragement. At the same time, this usage has never appeared in COCA[1] (Dong, 2020).

During the translation process, semantic

---

[1] Available at: `https://www.english-corpora.org/coca/`

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 59–69
November 8-9, 2025 ©2025 Association for Computational Linguistics



1	he trial judge noted that the operative intervention	caused	the **rupture** -- not the failure to earlier diagnose
2	inutes each and where even a slightest error would	cause	**death**. Because that would NOT be fun. Also, if the
3	omfort. They are specially designed to relieve **pain**	caused	by bruxing or clenching. # If you believe you grin
4	enters of power. They have spilled more blood and	caused	more **misery** than any other force. They have canoni
5	and Limn. # The drop in rainfall and soil saturation	caused	**landslides** in the canton of Turrialba, Paraiso, Ji
6	system cope with the impending public health crisis	caused	**Alzheimer's disease** and related dementia will be p
7	, recently elected Socialist Party leader Harlem Dsir	caused	**a stir** when he reminded the crowd of President Fra
8	SE by Vivian Sharpe. It explains the **damage** that is	caused	by sexual abuse. It reminds us why it is so import
9	d how autism is the the result of **" brain damage "**	caused	by vaccines; then the confabulation begins that re
10	search engines have her name blocked. **Diabetes** is	caused	by parasites? So, if she didn't have an accumulati
11	of Somalia have made the waters their home and	have caused	**extensive damage** as a result. Shipping industries
12	be videos. # And with **all the astrosities** this animal	has caused	, if he does not get impeached, tried. sentenced,

Figure 1: Concordance lines of *CAUSE* in the Corpus of Contemporary English in the United States (COCA).

prosody should be considered in order to achieve semantic/pragmatic equivalence and avoid missing information. Since it is difficult for individual non-native speakers of a language to correctly perceive semantic prosody with intuition, even professional human translators may fail to convey it into the target text, as they usually translate from their second language (L2) into their mother tongue. In both translation pedagogy and L2 learning, semantic prosody awareness has been considered to be important and of value to learners (Stewart, 2009; McGee, 2012).

Inequivalence of semantic prosody is also a problem for Machine Translation (MT), yet little attention has been paid to it. If we ask a model to translate "I was praised by my teacher" into Chinese, Google Translate, ChatGPT 4 and DeepL would give the following translations:

Google Translate & ChatGPT 4:

(1) 我 **被** 老师 表扬 了。
1SG BEI teacher praise PERF
'I was praised by my teacher.'

DeepL:

(2) 我 **受到** 了 老师的 表扬。
1SG SHOUDAO[2] PERF teacher's praise-VN
'I was praised by my teacher.'

These are two different literal translations in passive voice. In English, passive structures such as "be + past participle" (hereafter referred to as "*BE* passive") mainly occur in neutral contexts (Xiao et al., 2006). However, these structures have negative semantic prosody in Mandarin Chinese (Wu, 2022; Dong et al., 2023). As the standard and most common passive structure in Chinese, the "被 BEI + verb" structure (hereafter referred to as *BEI* passive) has obvious negative prosody, which is not an adequate option for translating a sentence talking about "being praised" or any other favourable situation.

To the best of our knowledge, no previous studies have attempted to teach an MT model about semantic prosody awareness. In this paper, we propose a method to incorporate semantic prosody information regarding a specific structure, namely BEI passive of Mandarin Chinese, into a Sequence-to-Sequence machine translation model. To this end, we introduce a dataset created to demonstrate the negative semantic prosody of BEI passives, which is later used to fine-tune MT models.

The main contributions of this paper are as follows:

1. We propose a method to teach semantic prosody awareness of a specific structure to Seq2Seq MT models, that is, fine-tuning them with a dataset that explicitly demonstrates the semantic prosody of the node.

2. We create a first-of-its-kind English-Chinese parallel dataset on semantic prosody. All sentence pairs are manually selected to illustrate the fact that Chinese BEI passive has negative semantic prosody and is the appropriate translation of English BE passives only if the context is unfavourable.

3. We employ a probing task to validate the idea that fine-tuned models contain more information that helps to decide whether a passive structure should be used in translation.

4. We achieve better performance with our fine-tuned models on translating passives while maintaining original BLEU, chrF2 and CometKiwi scores on Flores+ and Tatoeba datasets.

---

[2] SHOUDAO ("be given or undergo") is a delexicalized verb (light verb) and marks passive voice (Cai et al., 2019). In this light verb pattern, the main semantic content of the predicate is provided not by it, but by its action nominal complement, verbal noun 表扬 *BIAOYANG* ("praising"). The subject is the patient here.

2 Related Work

Research on semantic prosody has developed in multiple directions, ranging from theoretical discussions to empirical studies in contrastive linguistics and translation. In this section, we first outline the key concepts related to semantic prosody, and then examine how it has been applied in cross-linguistic studies and translation research.

2.1 Semantic Prosody

Semantic prosody is a collocational meaning formed through the co-occurrence of a node and a consistent series of collocates (Louw, 2000). Another related concept is semantic preference. A node can display its semantic preference by co-occurring with several items from a specific semantic set, which can contain favourable and unfavourable items at the same time. According to Partington (2004) semantic preference and semantic prosody have different operating scopes: the former relates the node item to another item from a particular semantic set whereas the latter can affect wider stretches of text. Semantic preference is a feature of the collocates, while semantic prosody is a feature of the node word. For example, *ABSOLUTELY* shows a semantic preference for words with a strong or superlative sense, such as *DELIGHTED, ENCHANTING, SPLENDID, PREPOSTEROUS, APPALLING, INTOLERABLE* (Partington, 1991). Both semantic prosody and semantic preference are established through collocates, and semantic preference "contributes powerfully" to building semantic prosody (Partington, 2004).

The definition of semantic prosody has not been undisputed over the last thirty years. Sardinha (2000) and Stubbs (2001) seem to consider that semantic prosody and connotation are synonymous, while Louw (2000) takes a different view. According to the *Collins Cobuild English Dictionary* definition of connotation, "the connotations of a particular word or name are the ideas or qualities which it makes you think of" (Sinclair, 1995). For example, *URCHIN* has a connotation of mischievousness. Louw (2000) takes connotation as "a form of schematic knowledge of repeatable events, e.g., what urchins do, where they live, their financial means or lack of it and how they behave, etc.". At the same time, semantic prosody is more contingent to collocates and requires corpus data to pin it down. It should be noted that although semantic prosody and connotation share the property of being attitudinal, the concealed quality is more fundamental to semantic prosody than it is to connotation (Stewart, 2010), which is far more accessible and can be learned in daily life.

2.2 Semantic Prosody in Contrastive Linguistics and Translation

In the field of contrastive linguistics, many case studies have been conducted on various language pairs of Indo-European languages and Sino-Tibetan languages. Xiao and McEnery (2006) compared the prosodies of near-synonyms across English and Chinese, and Sardinha (2000) analyzed English and Portuguese. Both studies conclude that the collocational behaviour and semantic prosodies of near-synonyms are unpredictable across the two language pairs, sometimes appearing similar and other times distinct. As for recent works, Wu and Lan's study on *BECAUSE* and 由于 *YOUYU* (Wu and Lan, 2019), and Dong's study about *INSIST ON* and 坚持 *JIANCHI* (Dong, 2020) also validate this observation. Partington (1998) claims that perfect equivalents across English and Italian are rare between because even words and expressions that are "look-alikes" (e.g., English *CORRECT* vs. Italian *CORRETTO*) may have very different lexical environments. Furthermore, there are many case studies discussing the more appropriate translation of a certain word or phrase (see example (3)). Wang and Ge (2021) claim that considering the negative semantic prosody of "It is what it is", (3-b) is a better translation than (3-a), since the former also has negative semantic prosody whereas the latter mainly appears in a neutral context.

(3) **Source text:** It is what it is.
  a. 情况就是这样。
     'This is the situation.'
  b. 事已至此。
     'The matter has come to this.'

Currently, although the application of semantic prosody to translation, translation pedagogy and L2 learning is the subject of research, yet no study has tried to appreciate it for improving MT performance. Considering its importance in translation equivalence, teaching models about semantic prosody awareness is a feasible way to improve MT translation performance.

3 Dataset Description

We propose teaching machine translation models about the negative semantic prosody of Chinese BEI passives by fine-tuning them with a dataset focusing on this structure, (hereafter referred to as the BEI dataset). Detailed information on the dataset is presented in this section.

3.1 Linguistic Structures

3.1.1 BEI Passives

The BEI dataset was created to explicitly demonstrate the negative semantic prosody of a passive structure in Chinese (*BEI* + verb) to a translation model. *BEI* is a grammatical passive marker without a concrete semantic meaning and is the most frequently used one among all passive markers. In Chinese, an active sentence can be turned into a passive one through adding a passive marker, and switching the subject and object, that is, the patient becomes the subject. Example (4) shows how such change is made.

(4) a. 张三 打 了 李四
Zhangsan beat PERF Lisi
'Zhangsan beat Lisi'

b. 李四**被** 张三 打 了
Lisi **BEI** Zhangsan beat PERF
'Lisi was beaten by Zhangsan'

The frequency, genre distribution, and semantic prosody of the passive voice differ in Chinese and English. Through an analysis of a corpus of recent materials (literature, news, and papers from January 1st to October 20th, 2021), Dong et al. (2023) reveals that the passive voice is approximately eight times more common in English than in Chinese. In English, it primarily conveys *neutral* content and is more frequent in news and academic articles, which require objectivity, than in novels. In contrast, Chinese uses the passive voice mainly for *negative* content, with little variation across genres, as illustrated in example (5).

(5) 家珍**被**拖出去时，双手紧紧捂着凸起的肚子，那里面有我的儿子呵。
'As Jiazhen **was carried out**, her hands firmly clasped her protruding belly, which held my son.'

The frequency of BEI passives in fiction is higher than in other genres, appearing 153 times per 100,000 words in literary texts, while only 94 times per 100,000 in news texts and even less frequently in scientific papers and miscellaneous texts. Meanwhile, the semantic prosody of BEI passives is also the most negative in literary texts, compared to that in other genres. In 66% of cases, the BEI passive has negative collocates, whereas the percentage in news text is 51.5% (Xiao et al., 2006). The frequency of passive sentences in Chinese translated fiction is lower and the semantic prosody of BEI passives is more negative than in Chinese original fiction, showing a tendency toward domestication in translation (Jia, 2010). Considering these facts, the dataset was created only with literary texts.

It should also be noted that although the passive voice is mainly used for unfavourable events in Chinese, its usage in a positive context is not entirely nonexistent. As with the active voice, it can be used in any context, positive and negative alike.

3.1.2 BE Passives

The structure *BE* + past participle can be considered the norm for English passives (Xiao et al., 2006) and is the most frequent passive structure used in English. BE passives appeared 9,908 times in FLOB (Freiburg-LOB corpus[3], an update of the Lancaster-Oslo-Bergen corpus of British English that contains texts published between 1991 and 1992), while GET passives appeared only 59 times. Thus, the BE passive is the structure we looked for when collecting sentence pairs.

English BE passives and Chinese BEI passives show great divergence in semantic prosody. According to Xiao and McEnery (2006), unlike BEI passives, 80% of BE passives in FLOB and BNCdemo (a demographic sampled component of the British National Corpus[4], the World edition) express neutral content.

3.2 Dataset Creation

Our dataset contains 900 English-Chinese parallel sentences manually selected from the fiction genre of The Babel English-Chinese Parallel Corpus (244,696 words in total) created by Richard Xiao, and from the China English-Chinese Parallel Corpus-Core (CECPC-Core, 5,499,591 words in total) created by Kefei Wang of BFSU [5].

---

[3] Available at: `https://clarino.uib.no/korpuskel/corpora`

[4] Available at: `http://www.natcorp.ox.ac.uk/`

[5] All available at the CQPweb of Beijing Foreign Studies University: `http://114.251.154.212/cqp/`.

3.3 Dataset Analysis

In the BEI dataset, the source texts contain BE passives in all persons and tenses. There are two subsets created with different requirements, namely positive evidence and negative evidence of the usage of BEI passives in the translation of English BE passives:

Positive evidence: 476 sentence pairs in which English BE passives are translated to Chinese BEI passives by a human translator and express negative content. They are selected to reinforce the relation between BEI passives and negativity.

Negative evidence: 424 English BE passives translated into Chinese with active voice by human translators, and the corresponding Chinese translation. This subset is intended to attenuate the degree of correspondence between the two passives, so that models may use more active voice when translating BE passives into Chinese.

4 Experimental Setup and Analysis

In this section, we describe our experiments teaching MT models about the negative semantic prosody of BEI passives. We also present the experimental setup for fine-tuning and probing, a comprehensive analysis of the evaluation and our results.

We experimented with three Sequence to Sequence (Seq2Seq) MT models: *Helsinki-NLP/opus-mt-en-zh* (Tiedemann and Thottingal, 2020; Tiedemann et al., 2023), *facebook/nllb-200-distilled-600M* (Costa-Jussà et al., 2022) and *facebook/mbart-large-50-many-to-many-mmt* (Tang et al., 2020)—hereinafter referred to as OPUS-MT, NLLB-600M and mBART50-mmt. All models reach state-of-the-art performance on English-Chinese text translation.

The evaluation was done in two parts. We tested the performance of the fine-tuned models for translating general text with the Flores+ (NLLB Team et al., 2024; Costa-Jussà et al., 2022) and Tatoeba (Tiedemann, 2020) datasets. BLEU (Papineni et al., 2002), chrF2 (Popović, 2017) and CometKiwi (Rei et al., 2022) metrics were used to evaluate the translation. After that, we used the BEI dataset test split to see whether the fine-tuned models had learned about the negative semantic prosody of BEI passives and if they had a higher accuracy deciding when to use BEI passives in translation.

Finally, a probing task was conducted to assess whether the pretrained and fine-tuned models possess information that aids in correctly using the active or passive voice when translating English passives into Chinese. The task also aimed to identify which layers of the encoder or decoder store this information.

4.1 Model Fine-tuning

We fine-tuned OPUS-MT, NLLB-600M and mBART50-mmt with the BEI dataset for English-Chinese text translation. The BEI dataset was split into 75% training, 11.25% validation and 13.75% test. We conducted a hyperparameter search for the learning rate and used batches of 32. For OPUS-MT, we trained with a learning rate of 10^{-5} and acquired the model with the best validation BLEU after 6 epochs. For NLLB-600M, we trained with a learning rate of 5×10^{-4} for 5 epochs, and kept the checkpoint with the best validation BLEU (at step 60). For mBART50-mmt, the appropriate learning rate is also 10^{-5}. We trained for 5 epochs and at step 120 we obtained the model with the best BLEU.

4.2 Model Performance Analysis

4.2.1 General Text Translation

In Table 1, we can see that the fine-tuned OPUS-MT achieved slightly higher BLEU and chrF2 scores on the Tatoeba dataset, fine-tuned NLLB-600M achieved a higher BLEU on Flores+ and a higher CometKiwi on Tatoeba, and fine-tuned mBART50-mmt achieved a higher CometKiwi on Tatoeba dataset. In general, after fine-tuning, the models maintained their original accuracy in the English-Chinese text translation task.

4.2.2 BEI Test Set Translation

The BEI test set contains 65 positive evidence sentence pairs and 59 negative evidence sentence pairs. The performance of pretrained and fine-tuned models is shown in Table 2. Generally speaking, all three models tend to use BEI passives to translate BE passives, with the positive evidence test set yielding higher accuracy than the negative evidence test set. All three models achieved higher accuracy in using BEI passives when translating BE passives after fine-tuning. For the positive evidence test set, fine-tuned mBART50-mmt shows the highest accuracy, which means it used BEI passives to translate BE passives with unfavourable content in most cases. For the negative evidence dataset, fine-tuned NLLB-600M performs the best,

Models	Flores+			Tatoeba		
	BLEU	chrF2	CometKiwi	BLEU	chrF2	CometKiwi
OPUS-MT	**32.1**	**21.6**	83.9	32.7	21.9	79.5
Fine-tuned OPUS-MT	31.7	21.3	**84.3**	**33.7**	**22.7**	**80.9**
NLLB-600M	23.3	**17.1**	**82.3**	**27.7**	**18.4**	69.7
Fine-tuned NLLB-600M	**24.4**	16.4	80.4	25.3	18.1	**74.7**
mBART50-mmt	**32.8**	**22.6**	**84.8**	**33.9**	**22.8**	81.6
Fine-tuned mBART50-mmt	32.3	21.4	84.7	33.2	22.4	**82.0**

Table 1: BLEU, chrF2 and CometKiwi scores (in %) of pretrained and fine-tuned models on Flores+ and Tatoeba datasets (English to Chinese). Fine-tuning leads to a minor drop in general translation quality, the models retain accuracy close to the original ones.

avoiding BEI passives in the translation of BE passives with neutral or favourable content.

Since in Chinese it is always acceptable to use the active voice for all kinds of content in Chinese, it is more important to achieve high accuracy in the negative evidence test set. Examples (6) and (7) show how fine-tuned models avoid using BEI passives when translating a BE passive stating a neutral or favourable event:

(6) **Source text:** Oh yes, and I **have been told** they played all sorts of mad pranks.

Target text: 有的。人家和我说，他们做了好多发疯似的把戏。

‘Yes. They **have told** me that they played many mad pranks.’

OPUS-MT: 哦，是的，我**被**告知他们玩各种疯狂的恶作剧。

‘Oh yes. I **have been told** that they played all sorts of mad pranks.’

Fine-tuned OPUS-MT: 哦,是的,有人告诉我，他们玩各种疯狂的恶作剧。

‘Oh yes. Someone **has told** me that they played all sorts of mad pranks.’

(7) **Source text:** You **were treated** as a son in my friend’s house.

Target text: 你在我朋友家里是待你同儿子一样的。

“In my friend’s house they **treated** you as a son.”

NLLB-600M: 在我朋友的家里, 你**被**当作儿子。

‘In my friend’s house, you **were treated** as a son.’

Fine-tuned NLLB-600M: 你在我朋友的家里就像一个儿子一样。

‘In my friend’s house you **were** like a son.’

4.2.3 Transference to Spanish-Chinese

Since NLLB-600M is a multilingual model, we hypothesize that the knowledge regarding the negative semantic prosody of BEI passives can be transferred and applied when translating passive sentences from other languages into Chinese. According to data from CORPES (*Corpus del Español del Siglo XXI*[6]), like in English, Spanish passive structures *SER*/*ESTAR* + past participle also have neutral semantic prosody (appearing in neutral contexts in around 78% of all cases). We had the BEI test set translated into Spanish by a native Spanish speaker and guaranteed that all the passive structures were preserved. Pretrained and fine-tuned NLLB-600M performance on the Flores+ and Tatoeba dataset (Spanish-Chinese), as well as BEI test set in Spanish, is shown in Table 3. Due to the unsolved issue of producing approximately 50% its output in Englsih when asked to translate Spanish to Chinese, mBART50-mmt performs poorly on this task and is not discussed here[7].

After fine-tuning with the English-Chinese BEI dataset, NLLB-600M achieved higher accuracy on both positive and negative evidence sentences in

---

[6] Available at: `https://www.rae.es/corpes/`

[7] Discussion raised at: `https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt/discussions/13`.

BEI test set	OPUS-MT	Fine-tuned OPUS-MT	NLLB-600M	Fine-tuned NLLB-600M	mBART50-mmt	Fine-tuned mBART50-mmt
positive evidence	75.4	81.5	69.2	78.5	81.5	**84.6**
negative evidence	10.2	25.4	35.6	**59.3**	35.6	50.8

Table 2: Accuracy (in %) of pretrained and fine-tuned models on BEI test set (English to Chinese).

Model	Flores+			Tatoeba			Spanish-Chinese BEI test set	
	BLEU	chrF2	CometKiwi	BLEU	chrF2	CometKiwi	pos-evi	neg-evi
NLLB-600M	16.8	**13.1**	**65.8**	**35.1**	**23.6**	**81.0**	66.2	28.8
Fine-tuned NLLB-600M	**17.9**	13.0	65.5	29.5	21.9	78.3	**80.0**	**59.3**

Table 3: Performance (in %) of pretrained and fine-tuned NLLB-600M model on Flores+, Tatoeba and the Spanish-Chinese BEI test set (transfer learning).

the BEI test set translated into Spanish. This indicates that knowledge of the negative semantic prosody of BEI passives learned from English-Chinese fine-tuning can be applied to other language pairs. As for the Spanish-Chinese translation performance, NLLB-600M maintained it on Flores+, but it worsened slightly on Tatoeba.

4.3 Probing Task

Probing classifiers have been used to measure and interpret how certain linguistic information is encoded in deep neural networks. The idea is to train a classifier (a probe) to predict a linguistic property from a model representation (Belinkov, 2022). In our case, we use probing to determine whether MT models store information in their representations, and whether such information helps to decide on the use of the passive voice in the Chinese translation of a BE passive.

The probe input consists of the hidden state of each layer of the models, including encoder layers and decoder layers. Our probing classifier has a single linear layer initialized with the same set of weights for all models and layers. The probing task is a simple classification task. We built a probing dataset with the validation and test set of the BEI dataset, which includes 225 BE passives, and a number indicating whether the translation should be an active or passive sentence (0 for active and 1 for passive). We used 80% of the probing dataset for training the probe, and the remaining 20% for testing probing accuracy. We trained a probe and recorded its accuracy on the probing test set for each layer of all three pretrained models and our fine-tuned models. This revealed which layer encodes the most information that could help models decide whether active or passive voice is more appropriate for the translation. The probing results are shown in Figure 2.

NLLB-600M and mBART50-mmt models have the same architecture. Both consist of a 12-layer encoder and a 12-layer decoder. In the encoders, we can observe that deeper layers tend to achieve higher accuracy on the probing task. Fine-tuning improved the mean accuracy of the encoder probes on mBART50-mmt by approximately 3%, but for NLLB-600M, it did not make a great difference. As for the decoders, it seems that the linguistic information that helps choose the correct voice in Chinese translation is distributed equally in all layers. The probes for the decoders of both models reached a higher mean accuracy after fine-tuning (an approximate 4.5% improvement). OPUS-MT has a 6-layer encoder and a 6-layer decoder, with a smaller layer size compared to the other two models[8]. In the encoder, fine-tuning had almost no effect on probing accuracy. We observe that layer 3 achieved a higher accuracy compared to all other layers. As for the decoder, fine-tuning caused layers 3 and 4 to contain less linguist information related to our task, while giving more information to the last layer.

Overall, with input from the NLLB-600M and mBART50-mmt layers, the probe shows a better ability to distinguish those BE passives that should be translated into Chinese active sentences from those that can keep their passive voice, whereas layers of OPUS-MT contain less information. This is in line with the performance of the models on the BEI test set and may suggest that models with more

---

[8] The hidden size of NLLB-600M and mBART50-mmt is 1024, while for OPUS-MT it is 512.

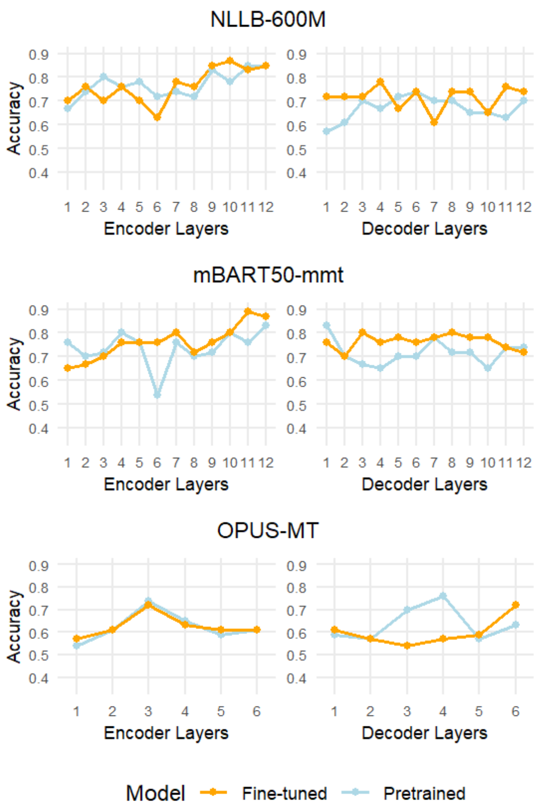


Figure 2: Layer-wise probing accuracy of pretrained and fine-tuned models. Encoder probes for NLLB-600M and mBART50-mmt show higher accuracy in deeper layers. Fine-tuning improves probe accuracy, particularly for the decoder layers. OPUS-MT encoders contain less task-relevant information overall, with accuracy peaking at layer 3.

layers have a higher ability to capture information related to semantic prosody.

5 Discussion

After fine-tuning the different models with the BEI dataset, they all showed improvements in reserving Chinese BEI passives for negative contexts. Here, we have a closer look at those cases with negative evidence that the model managed to translate correctly into the active voice, in order to find potential methods to further improve model performance.

The cases that models corrected after fine-tuning can be categorized into two kinds according to the translation strategy. For some BE passives, models switched the position of the subject and the object and removed the passive marker—such as in example (6)—in order to give a sentence in active voice. This is the kind of change we expected to see. The second translation strategy is to use a notional passive, which is also called a topic sentence, as in example (8):

(8) **Source text:** That they **have been preserved** so well.

Target text: 它们都保存得十分完整。

‘They are in perfect preservation.’

mBART50-mmt: 它们**被**保存得如此好。

‘They **have been preserved** so well.’

Fine-tuned mBART50-mmt: 它们保存得很好。

‘They are in very good preservation.’

In a topic sentence, the subject argument is the topic and the patient of the verb (*THEY* in example (8)), while the remaining constituent is the comment. Topic sentences can express passive meaning because of the nature of their subjects, and sometimes it is grammatical to add a passive marker and turn it into a marked passive sentence, depending mainly on the verb. However, notional passives are only compatible with a very restricted number of verbs, and are not considered passive constructions in a strict sense (Tang, 2003).

In modern Mandarin Chinese literature, notional passives mainly occur in neutral contexts (over 80% of the time) and are twice as frequent as BEI passives (Guo and Chow, 2013), which means it is a plausible choice for translating BE passives into Chinese. Moreover, in example (8) we can observe that translating a BE passive into a notional passive does not require the model to change the order of subject and verb or to add back the omitted agent of the verb. The fine-tuned model only had to drop *BEI* 被 between the patient *THEY* 它们 and verb *PRESERVE* 保存 to obtain a topic sentence. In this case, *THEY* 它们 refers to non-human arguments, which means *THEY* cannot conduct the action of preserving something, but can only be the patient of the verb *PRESERVE*. Without the need to add the passive marker *BEI*, the sentence already transfers the passive notion that *THEY* are *PRESERVED*.

Using topic sentences to translate BE passives requires less movement and can preserve the passive meaning and neutral semantic prosody at the same time. However, while the BEI passive is more frequent in translated Chinese literature than in original Chinese literature, the opposite is the case with the notional passive (Guo and Chow, 2013). Both human translators and machine translation models should pay more attention to this structure to generate more accurate Chinese translations for passive sentences.

6 Conclusion and Future Work

In this work, we propose an approach for teaching sequence to sequence machine translation models about the semantic prosody of a specific structure, namely the Chinese BEI passive, and to improve model performance on translating English BE passives into Chinese. The primary focus of our approach is to fine-tune MT models with a dataset that explicitly demonstrates the negative semantic prosody of BEI passives through the contrast with using BEI passives to translate BE passives with negative context, while for BE passives with neutral and positive content, the translations are in the active voice. After fine-tuning OPUS-MT, NLLB-600M and mBART50-mmt, all models showed improvements in using BEI passives correctly in translation while maintaining their original performance on general text translation, showing that our approach is a valid one. Moreover, for multilingual models such as NLLB-600M, the knowledge of semantic prosody is transferable to translation tasks in other language pairs (e.g. from English-Chinese translation to Spanish-Chinese translation).

Through probing experiments, we found that, for NLLB-600M and mBART50-mmt, information that helps to decide whether the active or passive voice is more plausible in translation is concentrated in the deeper layers of the encoder and equally distributed in the decoder. Fine-tuning improved the probing accuracy of decoders significantly, but did not have a great influence on the encoders.

Our work focused on the BEI structure. A potential future research direction will be fine-tuning a model with combined data that can show the semantic prosody of multiple linguistic units and observe whether the model can learn different semantic prosodies simultaneously. Having a more diverse dataset can also prevent the problem of overfitting. However, there is no comprehensive list of linguistic units from different languages with semantic prosody that may cause inequivalence in translation. It would be of great value to compile such a list and create datasets that demonstrate their semantic prosody.

In English and Spanish, passive structures can be categorized into two different kinds, namely adjectival and verbal passives. In this work, we did not distinguish between them, as they present the same problem when translated into Chinese by MT models. However, we plan to investigate further to determine whether model performance varies on different kinds of passives.

Finally, another avenue for future work would be multilingual fine-tuning, which may yield better results on multilingual models. The difficulty here lies in the limited numbers of annotated multilingual parallel corpora, without which it would be difficult to collect data on a specific structure.

Limitations

Although our dataset is enough to demonstrate the negative semantic prosody of Chinese BEI passives, there is another important fact that our dataset does not show. That is, the low frequency of passive sentences (including BEI passives and others) in general Chinese texts, which should also affect the usage of BEI passives in translation. Since all the models are pretrained models, they should have seen this fact in the training process already. Whether there is necessity of fine-tuning models with BEI dataset mixed in general text needs future work to justify.

Acknowledgements

This work was supported by FairTransNLP-Language: Analysing toxicity and stereotypes in language for unbiased, fair and transparent systems (PID2021-124361OB-C33), funded by Ministerio de Ciencia, Innovación y Universidades, programa de I+D de Generación de Conocimiento (MICIU/AEI/10.13039/501100011033/FEDER,UE) and by Support grants to departments and university research units for the recruitment of pre-doctoral research staff in training (FI SDUR 2024).

References

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Huiying Cai, Yunhua Qu, and Zhiwei Feng. 2019. A corpus-based study of the semantic prosody of chinese light verb pattern across registers: Taking jinxing and shoudao as examples. *Glottometrics*, 46:61–82.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Di Dong. 2020. A comparative research on the insist semantic prosodic phrases and pragmatic attributes based on corpus. *Innovation and Practice of Teaching Methods*, 3:68.

Pufan Dong, Caixia Jiang, and Pengyuan Xu. 2023. A corpus-based comparative study of english chinese passive voice. *Oversea English*, 16:53–55+59.

John Rupert Firth. 1964. Papers in linguistics, 1934-1951.

Hongjie Guo and Daryl Chow. 2013. A corpus-based variationist approach to bei passives in mandarin chinese. *Corpus Linguistics and Linguistic Theory*, 10(1):139–173.

Zenghong Jia. 2010. The frequency,structure and semantic prosody of ”bei” passives in chinese translated fiction. *Journal of foreign languages*.

Bill Louw. 1993. Irony in the text or insincerity in the writer? the diagnostic potential of semantic prosodies. *Text and technology: In honour of John Sinclair*, 157:176.

Bill Louw. 2000. Contextual prosodic theory: Bringing semantic prosodies to life. *Hef fer Chris, & Saunt son Helen (eds.)*, pages 48–94.

Tony McEnery, Richard Xiao, and Yukio Tono. 2006. *Corpus-based language studies: An advanced resource book*. Taylor & Francis.

Iain McGee. 2012. Should we teach semantic prosody awareness? *RELC Journal*, 43(2):169–186.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alan Partington. 1991. A corpus-based study of the collocational behavior of amplifying intensifiers in english. *Unpublished masters thesis). University of Birmingham, England*.

Alan Partington. 1998. *Patterns and meanings: using corpora for English language research and teaching*. John Benjamins.

Alan Partington. 2004. ” utterly content in each other’s company”: Semantic prosody and semantic preference. *International journal of corpus linguistics*, 9(1):131–156.

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

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC De Souza, Taisiya Glushkova, Duarte M Alves, Alon Lavie, and 1 others. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. *arXiv preprint arXiv:2209.06243*.

Tony Berber Sardinha. 2000. Semantic prosodies in english and portuguese: A contrastive study. *Cuadernos de filologia Inglesa*, 9(1).

John Sinclair. 1987. *Looking Up: An Account of the COBUILD Project in Lexical Computing and the Development of the Collins COBUILD English Language Dictionary*. Collins Cobuild Dictionaries Series. Collins ELT.

John Sinclair. 1995. *Collins Cobuild English dictionary*. Collins CoBUILD.

John Sinclair. 1996. The search for units of meaning. *Textus*, 9(1):75–106.

Dominic Stewart. 2009. Safeguarding the lexicogrammatical environment: Translating semantic prosody. In *Corpus Use and Translating: Corpus use for learning to translate and learning corpus use to translate*, pages 29–46. John Benjamins Publishing Company.

Dominic Stewart. 2010. *Semantic prosody: A critical evaluation*. Routledge.

Michael Stubbs. 1995. Collocations and semantic profiles: On the cause of the trouble with quantitative studies. *Functions of language*, 2(1):23–55.

Michael Stubbs. 2001. *Words and phrases: Corpus studies of lexical semantics*. John Wiley & Sons.

Sze-Wing Tang. 2003. Three syntactic issues of chinese passives. In *New Expansion of Research on Chinese Passive Expression Problems–Proceedings of the International Symposium on Chinese Passive Expression Problems*, pages 100–107.

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

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato Yves Scherrer, Raul Vazquez, and Sami Virpioja. 2023. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, (58):713–755.

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

Tiantian Wang and Shili Ge. 2021. Corpus-based semantic prosody study of english-chinese translation: taking trump' s popular saying “it is what it is” as an example. In *Learning Technologies and Systems: 19th International Conference on Web-Based Learning, ICWL 2020, and 5th International Symposium on Emerging Technologies for Education, SETE 2020, Ningbo, China, October 22–24, 2020, Proceedings 5*, pages 420–429. Springer.

Zhong Wu. 2022. A quantitative study on the stylistic differences of “bei” passives in contemporary chinese. In *Computational Social Science*, pages 50–55. Routledge.

Zhong Wu and Xi-Jun Lan. 2019. The semantic prosody of “youyu” : evidence from corpora. In *Workshop on Chinese Lexical Semantics*, pages 654–660. Springer.

Richard Xiao and Tony McEnery. 2006. Collocation, semantic prosody, and near synonymy: A cross-linguistic perspective. *Applied linguistics*, 27(1):103–129.

Richard Xiao, Tony McEnery, and Yufang Qian. 2006. Passive constructions in english and chinese: A corpus-based contrastive study. *Languages in contrast*, 6(1):109–149.

Semantic Differentiation in Speech Emotion Recognition: Insights from Descriptive and Expressive Speech Roles

Rongchen Guo[1*], **Vincent Francoeur**[2*], **Isar Nejadgholi**[1,3*],
Sylvain Gagnon[2], **Miodrag Bolic**[1†]
[1]School of Electrical Engineering and Computer Science, University of Ottawa
[2]School of Psychology, University of Ottawa
[3]National Research Council Canada, Ottawa, Canada
`{rongchen.guo, vfran022, sgagnon, Miodrag.Bolic}@uottawa.ca`
`isar.nejadgholi@nrc-cnrc.gc.ca`

Abstract

Speech Emotion Recognition (SER) is essential for improving human-computer interaction, yet its accuracy remains constrained by the complexity of emotional nuances in speech. In this study, we distinguish between *descriptive semantics*, which represents the contextual content of speech, and *expressive semantics*, which reflects the speaker's emotional state. After watching emotionally charged movie segments, we recorded audio clips of participants describing their experiences, along with the intended emotion tags for each clip, participants' self-rated emotional responses, and their valence/arousal scores. Through experiments, we show that descriptive semantics align with intended emotions, while expressive semantics correlate with evoked emotions. Our findings inform SER applications in human-AI interaction and pave the way for more context-aware AI systems.

1 Introduction

The ability to accurately detect and interpret emotions in speech is vital for developing intelligent systems capable of natural and empathetic human-computer interactions. Speech Emotion Recognition (SER) has gained significant traction in recent years, driven by applications ranging from virtual assistants to mental health monitoring (Ley et al., 2019; Rumpa et al., 2015). Despite these advancements, SER faces persistent challenges due to the complex and multi-dimensional nature of emotions, which often intertwine with contextual and speaker-specific factors.

Traditional approaches to SER have largely focused on acoustic features, such as pitch, energy, and spectral properties, to infer emotional states (Wu et al., 2011; Bitouk et al., 2010; Venkataramanan and Rajamohan, 2019; Likitha et al., 2017; Kwon et al., 2003). While effective to some extent, these methods often overlook the semantic content of speech, which can provide crucial contextual information. With the advances in natural language processing, it has become increasingly feasible to analyze the semantic aspects of speech for emotion recognition (Tzirakis et al., 2021; Xu et al., 2021). However, the interplay between semantic roles and emotional expression remains underexplored. Specifically, the distinction between *intended emotions* elicited by a stimulus and *evoked emotions* experienced by the speaker is rarely addressed, leaving a critical gap in the field.

This paper introduces a novel framework to address this gap by distinguishing two types of semantic roles in speech. We hypothesize that *Descriptive semantics* captures scenario-specific content, such as the narrative or context described in the speech. In contrast, *Expressive semantics* reflects the speaker's subjective emotional stance, shaped by their personal experiences and delivery style. In our framework, descriptive segments are expected to align with the intended emotion of the stimulus (the target emotion the video was designed to elicit), while expressive segments are expected to align with the evoked emotion (the participant's self-reported experience). This mapping allows us to distinguish stimulus-driven affect from speaker-specific affect, thereby addressing a critical gap in prior SER research that often assumes a single ground-truth label. This semantic distinction is particularly important in settings where it is essential to understand not only *what happened* — the contextual content of speech — but also *how it was felt* — the speaker's emotional state and tone. Such an understanding has practical implications for applications like emotion-aware AI systems, educational tools, and interactive entertainment, where both the content and emotional delivery of speech play key roles in creating engaging and effective

---

[*] Equal contributions
[†] Corresponding author

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 70–82
November 8-9, 2025 ©2025 Association for Computational Linguistics

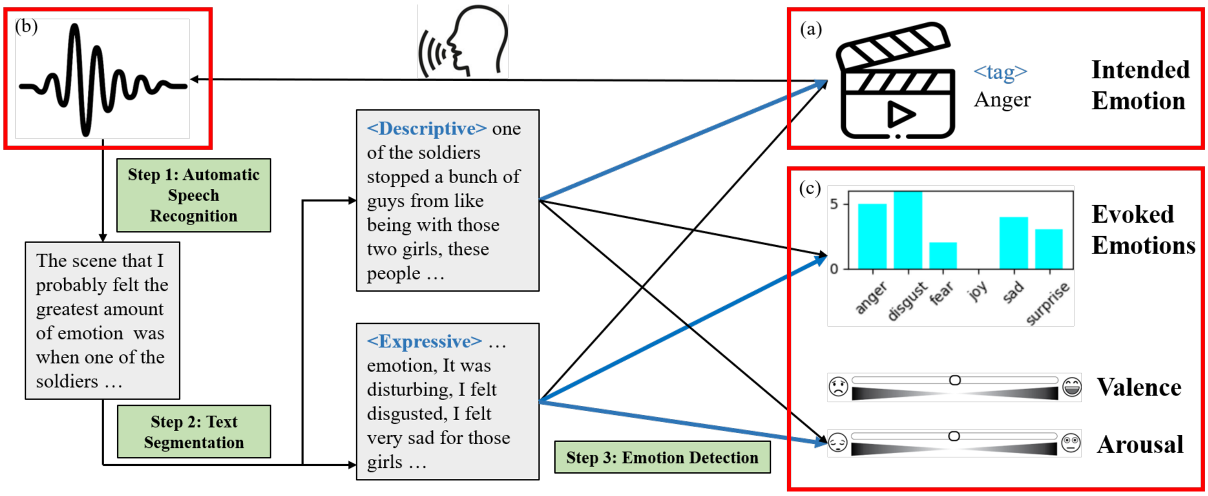

(b)
Step 1: Automatic Speech Recognition
The scene that I probably felt the greatest amount of emotion was when one of the soldiers …
Step 2: Text Segmentation
<Descriptive> one of the soldiers stopped a bunch of guys from like being with those two girls, these people …
<Expressive> … emotion, It was disturbing, I felt disgusted, I felt very sad for those girls …
Step 3: Emotion Detection
(a)
<tag>
Anger
Intended Emotion
(c)
5
0
anger
disgust
fear
joy
sad
surprise
Evoked Emotions
Valence
Arousal

Figure 1: **Data Collection and Algorithm Workflow:** Participants watched six videos eliciting specific emotions and provided speech descriptions, emotion ratings, and valence/arousal scores. Speech data were transcribed, segmented into **descriptive** and **expressive** semantics, and used to train models for three tasks: predicting intended emotions (TASK-1), evoked emotions (TASK-2), and valence/arousal (TASK-3).

human-computer interactions.

To validate our hypothesis, we collected a dataset comprising emotionally evocative movie clips to elicit a specific emotion. Participants watched the videos and provided ratings for the actual evoked emotions, alongside valence and arousal scores, creating a robust foundation for analysis. Our methodology to uncover the distinct relationships between semantic roles and intended versus evoked emotions involves three key steps: speech transcription with automatic speech recognition (ASR), semantic segmentation with LLMs, and emotion prediction with fine-tuned text classifiers/regressors. This work makes the following contributions:

- First, we curated a SER dataset with 582 audio recordings spanning six emotion categories. Audio transcriptions are generated, and intended emotions, as well as evoked emotions, are measured in an experimental setup.
- Second, we implemented an LLM-based semantic segmentation approach to separate the expressive and descriptive parts of speech and validated that through human evaluation.
- Third, through experimentation, we show that descriptive semantics are more predictive of intended emotions, while expressive semantics are better aligned with evoked emotions.

Importantly, our work goes beyond simply predicting emotion labels from participants' descriptions. By explicitly segmenting speech into descriptive and expressive roles, we quantify how different semantic functions relate to stimulus-intended versus self-experienced emotions. This role-based separation provides a principled way to reconcile discrepancies between intended and evoked affect and offers interpretable insights that are not available from standard text-only or audio-only models.

Our findings have significant implications for designing more accurate and context-aware emotion recognition systems, with potential applications in virtual assistants, customer service, and mental health support. By bridging the gap between semantics and emotion, this research advances the state-of-the-art in SER and sets the stage for future exploration of semantic roles in emotional AI systems.

2 Related Work

Emotions are complex psychological and physiological responses to salient events, involving bodily sensations, expressive behaviors, and cognitive evaluations (Moors, 2024, 2009). Various linguistic features, including prosody, lexical choice, and sentence structure, play a role in the perception and expression of emotions (Mohammad and Turney, 2010; Barrett et al., 2007; Keltner et al., 2019). Speech emotion recognition (SER) models aim to detect emotional states from speech using acoustic, textual, or multimodal signals. With the advancement of LLMs and automatic speech recognitions (ASR), text-based emotion classification has seen improved accuracy (Hama et al., 2024; Bekmanova et al., 2022; Bharti et al., 2022). Acoustic-based emotion detectors have also progressed using acoustic feature extractors, such as openSMILE (Eyben et al., 2010) or audio embedding models, such as wav2vec (Baevski et al., 2020) and HuBERT (Hsu et al., 2021), which embed paralinguistic cues such as pitch, tempo, and energy into speech representations (Ulgen et al., 2024; Chakhtouna et al., 2024; Zhao et al., 2024; Dutta and Ganapathy, 2024; Ghosh et al., 2016). Multi-modal approaches, which combine speech, facial expressions, and physiological signals, have also become increasingly prominent in recent years (Cheng et al., 2024; Khan et al., 2024; Morency and Baltrušaitis, 2017; Yoon et al., 2018; Niu et al., 2023).

Emotion elicitation via multimedia stimuli (e.g. short film clips) is a common technique in SER to induce targeted emotions (e.g., sadness, joy, fear) (Li et al., 2021; Rumpa et al., 2015). These movie-based emotion elicitation techniques have applications in various fields, including e-health monitoring and human-computer interaction (Ley et al., 2019; Rumpa et al., 2015). The stimuli are selected and validated through self-report and physiological measures (Chen et al., 2021; Handayani et al., 2015; Soleymani et al., 2012). While these methods control for the *intended* emotional target, they do not always account for the *evoked* emotion the speaker experiences and expresses. Prior work such as Siedlecka and Denson (2019) reviewed these paradigms in detail, but focused primarily on affect induction rather than the emotional content of participants' verbal responses. In our work, we analyze speech collected after stimulus exposure and study how intended and evoked emotions are reflected in participants' spoken descriptions. In doing so, we explore a novel distinction between semantic roles in language—namely, whether a speaker is being descriptive (e.g., summarizing the movie) or expressive (e.g., conveying their own reaction)—and how these roles align with different emotion types.

Many SER datasets have been developed. In acted speech datasets, such as IEMOCAP (Busso et al., 2008) and SAVEE (Jackson and Haq, 2014), actors are recruited to read sentences or act in scenes that portray different emotions. In spontaneous speech datasets, such as MSP-Podcast (Lotfian and Busso, 2017), MSP-Conversation (Martinez-Lucas et al., 2020), SAMAINE (McKeown et al., 2011), and RECOLA (Ringeval et al., 2013), and elicited speech datasets, such as LSSED (Fan et al., 2021), BAUM-1 (Zhalehpour et al., 2016), and eNTERFACE (Batliner et al., 2006), audios are recorded in a freely speaking environment or with emotion elicitation methods. Speech is then annotated by a third party (perception-of-other). However, these datasets focus on one emotion label per speech and do not distinguish different types of emotions. To this end, EMO-DB (Burkhardt et al., 2005) and IEMOCAP (Busso et al., 2008) analyzed emotional evocative sentences and perception-of-other in acted speech. Most similar to us, MuSE (Jaiswal and Bara, 2020) collects speech following emotional video stimuli and reports both self-reported and intended emotion annotations. While similar in structure, our work uniquely interprets the relationship between stimulus-intended and self-reported emotions through a *semantic* lens, enabling direct analysis of misalignment between the two emotion types.

Furthermore, some recent studies in NLP have explored emotion elicitation and manipulation in conversational settings (Gong et al., 2023; Ma et al., 2025; Qian et al., 2023; Meng et al., 2024). While our study does not model conversational interactions, our semantic framework may offer insights into these settings by helping to identify when emotional influence is being attempted or received. For example, expressive speech segments may signal internal affective states, while descriptive segments may reflect contextual awareness or narrative framing. These distinctions could inform models of emotion transfer and regulation in human-computer dialogue.

Our contribution lies in bridging the gap between stimulus-based emotion elicitation and the actual emotions conveyed by participants in speech. By segmenting utterances according to their semantic roles and analyzing how different roles align with either intended or evoked emotions, we propose a novel way to interpret emotional speech beyond traditional modality-based or label-based approaches. While prior SER studies have emphasized either acoustic or multimodal representations, our work suggests that semantic structure in language - accessible only through text - offers a distinct and interpretable signal for differentiating between types of emotion.

Movie Clip	Tag	Scene Description	Duration	Validation Source
The Blair Witch Project (Myrick et al., 1999)	Fear	Final scene when screaming intensifies, man standing facing the wall and camera falls.	2:03	Schaefer et al. (2010)
The Conjuring (Wan, 2013)	Fear	Girl gets out of bed at night and bags her head on a cupboard. Frantic scene.	2:26	İyilikci et al. (2024)
American History X (Kaye, 1998)	Anger	Neo-Nazi kills a black man, smashing his head on the curb and then smiles after being arrested.	3:24	Schaefer et al. (2010)
Platoon (Stone, 1986)	Anger	Villagers pushed around in burning village and soldier stops other soldiers from raping a child.	2:42	Author tested in pilot.
Baby laughing at ripping paper (YouTube, 2011)	Joy	8-month-old Micah (a boy) laughing hysterically while at-home daddy rips up a job rejection letter.	1:44	Author tested in pilot.
Cats and Dog playing together (YouTube, 2022a)	Joy	Dog lies peacefully on a large bed with kittens and adult cat moving around. With happy music.	1:53	Author tested in pilot.
One Day (Scherfig, 2011)	Surprise	Woman rides a bicycle; she gets hit by a truck.	2:26	Zupan and Eskritt (2020)
Neighbors (Nicholas Stoller and O'Brien, 2014)	Surprise	Woman calls man about missing airbags Man is ejected to an office ceiling.	1:07	Author tested in pilot.
Trainspotting (Boyle, 1996)	Disgust	The main character enters "The worst toilet in Scotland" and later dives into a filthy toilet bowl.	1:23	Schaefer et al. (2010)
Planet Terror (Rodriguez, 2007)	Disgust	Scene where man is examined by doctors in a hospital and exposes infected parts of his body.	2:01	Michelini et al. (2019)
Young impala and dead mother (YouTube, 2022b)	Sadness	Young impala finds adult impala lying down and apparently dead. Then lies by dead animal.	1:44	Author tested in pilot.
My Girl (Zieff and Elehwany, 1991)	Sadness	Funeral scene where girl cries and runs away after approaching the casket where a little boy lies.	1:39	Gabert-Quillen et al. (2015)

Table 1: Listing and information about the 12 movie clips used to elicit discrete emotions in the main study.

3 Dataset

The block diagram in Figure 1 summarizes our data collection, task definitions, and methodology, which we will elaborate on here and in the next section. Data collection was carried out in person at INSPIRE Laboratory of the School of Psychology at University of Ottawa. The experiment procedure was approved by the Research Ethics Board of University of Ottawa. The study included 97 student participants aged 18 to 27 (M = 19.9, SD = 2.5). The majority were women (81 women, 15 men, and 1 non-binary), and most participants were native English speakers (65 spoke English as their first language, 12 spoke French, and 20 spoke other languages). The sample was ethnically diverse, comprising 16 Asian, 20 Black/African, 7 Hispanic/Latino, 1 Indigenous, 15 Mixed/Multiple Ethnicities, 33 White/Caucasian, and 5 participants identifying as Other.

Our study focused on the six basic emotions identified by Ekman (1992) as the target emotions in our experimental setup: sadness, fear, joy, disgust, surprise, and anger. Two movie clips for each emotion were sourced from film stimuli in the existing literature and validated in our pilot study. The twelve movie clips used in the study and their meta-information are listed in Table 1. We trimmed clips to ensure optimal emotional salience and duration. Their effectiveness was validated in a pilot study with 25 participants before the final data collection.

In the main study, participants watched six emotional video clips, one from each emotion category. To re-establish baseline levels of valence and arousal, the presentation of each emotional clip was preceded by a neutral video clip. To further mitigate potential carryover effects between conditions, a two-minute rest period was inserted between each neutral–emotional clip sequence, during which one of six still images was displayed on the computer screen. All video clips and still images were presented in random order to minimize potential sequence effects. The collected dataset consists of 97× 6 entries, with five elements: 1) **Speech**: a 30-second audio recording of the participant's verbal response to the following instruction:

"You are asked to verbally describe the scene during which you felt the strongest emotion in the last film clip and say how it made you feel." 2) **Intended emotions**: Each video is expected to provoke a certain emotion. 3) **Evoked emotions**: the intensities at which each of the emotions (sadness, fear, joy, disgust, surprise, anger) was felt, as rated by the participants on a 7-point Likert scale going from *not at all* to *strongly*. 4) **Valence**: the extent to which the overall feeling of the participant was positive or negative. 5) **Arousal**: the intensity of the overall feeling of the participant while watching the video. Valence and arousal were measured on a validated sliding scale where each extreme was illustrated by an emoticon.

4 Tasks

We define three tasks to examine the relationship between semantic types and emotion recognition. To determine the most predictive semantic type for each task, we experimented with three different inputs: full transcriptions, descriptive semantic segments, and expressive semantic segments.

TASK-1: Classification of Intended Emotion involves classifying the intended emotion associated with each video based on participants' speech.

TASK-2: Classification of Evoked Emotion involves classifying participant-reported evoked emotions, which are subjective and may include multiple emotions simultaneously. While evoked emotions often include the intended emotion, individual differences can lead to variations. This task is framed as a multi-label classification problem, where each emotion (on a scale of 0 to 6) is binarized based on whether it is evoked or not.

TASK-3: Regression of Valence and Arousal predicts participants' self-reported valence and arousal ratings, which provide a two-dimensional representation of emotional states.

5 Methodology

As depicted in Figure 1, our methodology consists of three sequential steps: speech recognition, semantic segmentation, and emotion prediction.

Step-1: Automatic Speech Recognition - We used Whisper (Radford et al., 2023), an automatic speech recognition model, to transcribe the participants' speech data into text.

	Descriptive semantics	Expressive semantics
LLM & Annotator 1	0.71	0.73
LLM & Annotator 2	0.84	0.83
Annotator 1 & Annotator 2	0.77	0.74
Random & Random	0.63	0.64

Table 2: Human evaluations of GPT-4o text segmentations. The agreement between two human annotators was comparable to human-LLM agreements.

Step-2: Semantic Segmentation - We used GPT-4o (OpenAI, 2023) to extract descriptive and expressive segments from the transcription obtained in step 1. The prompt is given in Table 3. We set the sampling temperature to 0 to make the process more deterministic. Overlapping phrases were allowed when semantic roles intersected, ensuring comprehensive representation.

Step-3: Emotion Prediction - The last step is to perform tasks described in Section 4 to study the relationship between semantic roles and emotion recognition. Each model is trained and evaluated on three input types: full transcriptions, descriptive segments, and expressive segments.

Audio-Based Emotion Classification - In addition to text-based models, we also experimented with audio-based models trained directly on the speech recordings. These included a HuBERT model (Hsu et al., 2021), a Wav2Vec2 model (Baevski et al., 2020), and a baseline MFCC (mel-frequency cepstral features) classifier. The audio classifiers were evaluated on TASK-1 and TASK-2 using the full utterance audio as input. However, all speech-based models performed significantly worse than text-based classifiers. Since semantic role segmentation (i.e., distinguishing between descriptive and expressive segments) is inherently a linguistic task and not inferable from acoustic signals alone, we prioritized text-based methods for the core analyses of this paper moving forward.

6 Experiments

6.1 Implementation Details

For Step-1, automatic speech recognition, we used 'whisper-large-V3' [1], a state-of-the-art system ASR model known for its robustness across diverse accents and noise conditions. We manually reviewed transcriptions in the development set, consisting of 33×6 audio transcriptions. Whisper achieved a

---

[1] https://huggingface.co/openai/whisper-large-v3

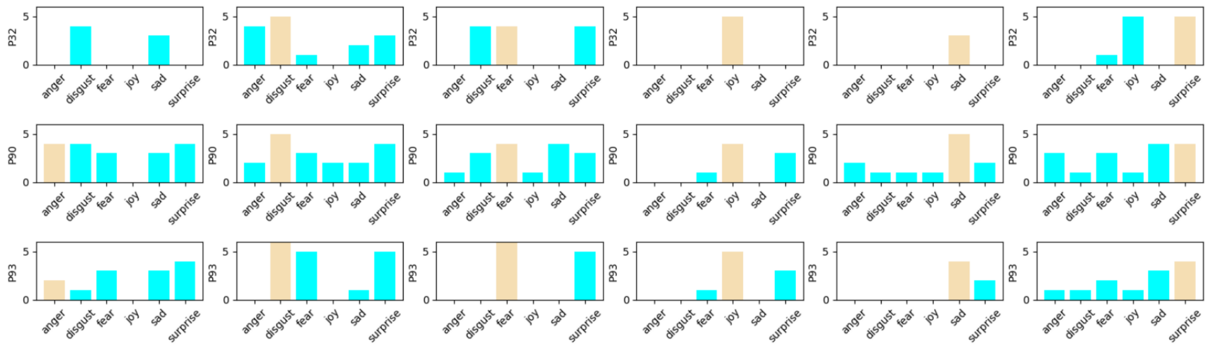


Figure 2: Examples of participants' rated emotions. Each row represents a participant who watched six movie segments (6 columns) from each of the six emotional categories. The intended emotion tag associated with the video is plotted in a yellow bar. Other rated emotions are colored blue. The height of the bars represents the emotion ratings from participants. For example, in the second movie clip watched by Participant P93, the intended emotion was "disgust," as shown by the yellow bar. After watching the clip, P93 reported experiencing four emotions: disgust, fear, sadness, and surprise, indicated by the blue bars. Among these, "disgust" was the strongest emotion, receiving the highest score of 6.

The user will provide a paragraph describing their feelings towards a particular movie, delimited with "'####"'.

Your task is to segment the paragraph into two parts according to the type of content: descriptive segments and expressive segments.

Descriptive segments refer to elements or clauses that provide factual or narrative information about the movie content without explicitly reflecting personal emotions or opinions.

Expressive segments refer to elements or clauses that convey personal feelings, attitudes, or opinions. These segments reflect individual reactions, emotions, and perceptions, or the intensity of these emotions.

The two parts (descriptive segments and expressive segments) can overlap, but all clauses of the given paragraph must be contained in at least one of the two parts.

Output your answer in the following format:

```
<answer>
  <descriptive> [descriptive segments] </descriptive>
  <expressive> [expressive segments] </expressive>
</answer>
```

Table 3: Prompt for extracting descriptive and expressive semantics from speech transcription.

4.13% word error rate, with errors mainly in unclear utterances at speech boundaries and between clauses.

For Step-2, we validated the effectiveness of GPT-4o segmentation again on the development set with 33 × 6 transcriptions. Two authors of this paper, one from the Computer Science department and the other from the Psychology department, were given the same instructions as the LLM and independently performed the same segmentation task. To calculate the agreement between human annotators and also between LLM and annotators, we computed cosine similarities of the segments, using sentence-transformer embeddings [2]. From Table 2, the average agreement between two human annotators (0.76) was comparable to human-LLM agreement (0.73 and 0.83). Most discrepancies arose from minor conjunctions to make sentences more complete. As a baseline, two random text segmentations would result in a similarity score of 0.63 - 0.64. Overall, GPT-4o has an acceptable segmentation quality.

For Step-3, emotion prediction, we fine-tuned different classifiers/regressors, including BERT (Devlin, 2018), RoBERTa (Liu, 2019), and DeBERTa (He et al., 2020). Different text semantics identified in Step-2 are used as inputs to the models. For emotion classification (Tasks 1 and 2), we used the text embeddings from the models and applied a standard classification head with a softmax activation function to predict categorical emotions. For regression (Task 3), we modified the models by replacing the classification head with a fully connected layer that outputs a single continuous value, trained with mean squared error (MSE) loss to predict valence and arousal scores. This approach follows standard practice in adapting transformer encoders for regression tasks (Xin et al., 2021; Taha, 2024; Orso and Xie, 2008). Data are split on participants' level, with 1/3 of partici-

---

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

How often do participants experience the intended emotion conveyed by the videos?	96.63%
How frequently do participants feel emotions other than the intended one?	89.39%
How often is the intended emotion rated as the highest by participants?	79.29%
Chippendale's alpha coefficientbetween intended emotion and evoked emotions	0.1466

Table 4: Statistics of relationships between movie intended emotion tags and evoked emotions. Predicting the evoked emotions is a much more subjective task than predicting the intended emotion tag.

pants (33 participants) data used for training, 1/3 for validation, and the rest 1/3 for testing.

6.2 Comparative Analysis of Intended and Evoked Emotions

Table 4 shows the relationship between the intended and evoked emotions. While the participants experienced the intended emotion 96.63% of the time, they also reported other emotions 89.39% of the time. Surprisingly, more than 20% of the time, an emotion other than the intended one is experienced most. These results suggest that the experienced emotion is highly subjective and can deviate from the intended emotions.

To better quantify the subjectivity of Task-2, we calculate the Krippendorff's alpha coefficient (Krippendorff, 2011, 2018) between the intended emotion and evoked emotions. We treat the agreement between intended emotion and evoked emotion as the agreement between two annotators performing multi-label annotations. Each annotator labels 594 data points, since there are 99 × 6 speech. One annotator always label the intended emotion as *true* and other emotions as *false*. The other annotator labels the data with the participant's evoked emotion ratings in a multi-label fashion. Krippendorff's alpha coefficient is calculated as the inter-annotator agreement index on this multi-label annotation task with MASI distance (Passonneau, 2006) as the distance measurement between two sets of multi-label annotations. The low score of Krippendorff's alpha coefficient shows the high subjectivity of task T2 and the high variation of evoked emotions with respect to the intended emotion.

Figure 2 gives examples of emotion ratings by three different participants in response to the six movie segments. Each row in the grid represents data from a different participant, while each column corresponds to one of the six movie segments. Within the bar charts, yellow bars indicate the intended emotion that the video clip aimed to elicit, while blue bars represent the emotions self-reported by the participants after watching the clips. The height of the bars reflects the intensity of the rated emotions on a numerical scale. These examples highlight variability in participants' emotional responses, often revealing discrepancies between the intended emotions and the emotions participants actually experienced.

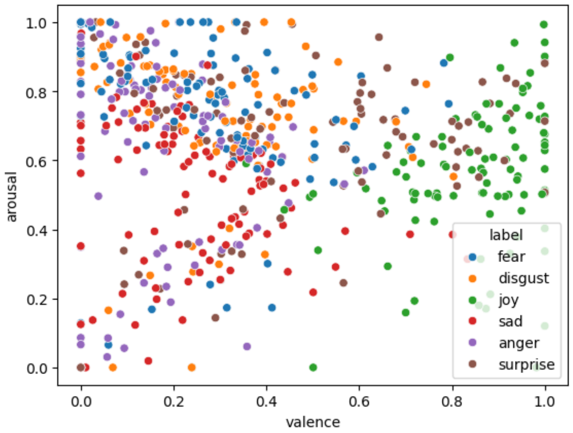


Figure 3: Valence and arousal ratings, colored by the intended emotion tags of movie segments.

6.3 Classification of Intended Emotion

Aligned with our hypothesis, the classification results for TASK-1 demonstrate a clear advantage of using descriptive semantics as input for predicting the intended emotions associated with each movie segment. Table 5 shows the classification accuracy for both semantic types across three different classifiers. Across all models, descriptive semantics consistently yield significantly higher accuracy in predicting the intended emotions.

6.4 Classification of Evoked Emotion

In TASK-2, we classified participant-reported evoked emotions, which are inherently subjective and may include multiple emotions simultaneously. Aligning with our hypothesis that expressive semantics better capture speaker-specific emotional experiences, results in Table 6 indicate that using expressive semantics as input achieves higher clas-

Model	Semantics	Precision	Recall	F1
BERT	Descriptive	0.83	0.81	0.81
	Expressive	0.68	0.65	0.65
	Full	0.89	0.88	0.88
RoBERTa	Descriptive	0.85	0.83	0.83
	Expressive	0.69	0.69	0.69
	Full	0.93	0.93	0.93
DeBERTa	Descriptive	0.81	0.81	0.81
	Expressive	0.65	0.64	0.64
	Full	0.91	0.90	0.90

Table 5: Model performances on classifying intended emotion associated with the movies.

Model	Semantics	Precision	Recall	F1
BERT	Descriptive	0.72	0.77	0.73
	Expressive	0.75	0.77	0.75
	Full	0.78	0.82	0.78
RoBERTa	Descriptive	0.73	0.76	0.73
	Expressive	0.74	0.82	0.77
	Full	0.76	0.82	0.77
DeBERTa	Descriptive	0.71	0.76	0.72
	Expressive	0.74	0.81	0.76
	Full	0.76	0.81	0.77

Table 6: Average model performances on classifying evoked emotions (std is always less than 0.1 over 5 run).

sification accuracy compared to using descriptive semantics. We also observe that even with full semantics, TASK-2 achieves significantly lower F-scores than Task-1, as expected due to the subjectivity of this task.

6.5 Discussion of Audio-Based Classifications

To assess the role of acoustic features in emotion recognition, we trained several audio-only classifiers, including models based on HuBERT [3], Wav2Vec2 [4], and MFCC features, for both TASK-1 and TASK-2. Across all models, we observed consistently poor performance, with classifiers frequently defaulting to one or two majority emotion classes. This suggests that prosodic and paralinguistic cues in our dataset were not strongly indicative of emotional content. One likely explanation is that participants generally delivered their responses in a steady and emotionally neutral tone, which limited the expressiveness of acoustic features.

Moreover, unlike text-based inputs, speech signals do not easily lend themselves to semantic segmentation without speech recognition (Wang et al., 2003; Ong and Herrera, 2005). Audio-based classifiers cannot distinguish between descriptive and expressive segments in an obvious way, making it difficult to explore the semantic roles that are central to our research questions. While acoustic features are valuable in many speech emotion recognition tasks, in our study design where subjective emotional experience is linked to semantic framing, textual cues proved more informative and interpretable.

6.6 Regression of Emotion Valence and Arousal:

Figure 3 shows the distributions of valence and arousal across different intended emotions, which exhibit high variability without clear patterns across emotions. Positive emotions, such as joy, correlates with higher valence, and negative emotions, such as fear, have lower valence and higher arousal. But there is no obvious clusters among the six emotions.

The results reported in Table 7 show that expressive semantics lead to more accurate predictions for both emotional valence and arousal compared to descriptive semantics. A statistical analysis in Table 8 shows that the differences in the prediction

---

[3] facebook/hubert-base-ls960

[4] facebook/wav2vec2-base

Model	**Semantics**	**Valence MSE**	**Valence MAE**	**Arousal MSE**	**Arousal MAE**
BERT	Descriptive	0.068	0.209	0.057	0.192
	Expressive	0.055	0.183	0.054	0.187
	Full	0.053	0.185	0.053	0.184
RoBERTa	Descriptive	0.050	0.184	0.055	0.184
	Expressive	0.037	0.151	0.051	0.182
	Full	0.034	0.146	0.051	0.182
DeBERTa	Descriptive	0.077	0.224	0.053	0.183
	Expressive	0.037	0.153	0.045	0.166
	Full	0.059	0.192	0.049	0.172

Table 7: Model performances on regression of emotion valence and arousal. Expressive semantics leads to smaller errors in estimating evoked valence and arousal. The difference is most pronounced for the DeBERTa-based model.

	Valence		Arousal	
Model	Z	p	Z	p
BERT	-1.74^a	0.083	-1.24^a	0.215
RoBERTa	-2.98^a	0.003	-0.16^a	0.874
DeBERTa	-5.86^a	< 0.001	-3.70^a	< 0.001

Table 8: Wilcoxon signed-rank tests results to compare MSE between descriptive and expressive semantics for each model. ^{a}Based on positive ranks.

errors between descriptive and expressive semantics are statistically significant for valence under two of the three models and one model for arousal. The regression results are in line with the TASK-2 results and the statistical analysis partially supports the hypothesis that expressive semantics better capture subjective experience.

7 Conclusion

This study introduces a novel framework for Speech Emotion Recognition (SER) by distinguishing between semantic roles in speech. By leveraging LLMs' zero-shot capabilities in text segmentation, we tackle a previously difficult challenge. To our knowledge, this is the first work to segment speech into two semantic roles, *expressive* and *descriptive* content, to enable more fine-grained and nuanced emotion detection.

Our findings reveal that descriptive semantics are more predictive of intended emotions, while expressive semantics are more closely aligned with evoked emotions and their valence and arousal dimensions. This differentiation can inform future research in emotion detection. In some contexts, it might be more useful to instruct users and guide them toward only one of these modes of expressing emotions. In other applications, it might be more suitable to leave it to the users to express their emotions in a mixture of expressive and descriptive modes. The LLMs can then be used to segment the speech and use the segments depending on the predictive goals. This approach enhances the development of more accurate and context-aware emotion recognition systems, with applications in mental health, virtual assistants, and customer service.

Limitations

This study, while providing valuable insights into the segmentation of speech for emotion recognition, has limitations. First, the dataset used in this research is curated from emotionally evocative movie clips, which, although varied, may not fully represent the broad diversity of real-world speech interactions. The emotional expressions captured in these clips might not encompass the full spectrum of spontaneous and everyday speech, which could limit the generalizability of the findings.

Second, although we included baseline speech-based emotion classifiers, their performance was substantially lower than that of text-based models. This gap likely stems from the emotional neutrality of the participants' tone and the nature of the task. However, future work could explore whether jointly modeling text and acoustic features, perhaps guided by semantic segmentation, might uncover latent prosodic patterns aligned with specific semantic roles.

Third, while the study distinguishes between descriptive and expressive semantics, it focuses primarily on self-reported emotional responses, which can be subjective and influenced by individual differences in emotional expression and perception. This subjectivity introduces variability in the emotional ratings, potentially affecting the accuracy and robustness of the regression models.

Ethics Statement

This research was approved by the Research Ethics Board of University of Ottawa. All participants provided their informed consent prior to participating in the study. Participants had the option to withdraw from the experiment at any time and for any reason, including emotional distress. Data collected during the study were handled securely and used exclusively for research purposes. All personal data was anonymized.

In Ethics Sheet for Automatic Emotion Recognition and Sentiment Analysis, Mohammad (2022) provides a structured ethical framework for developing and deploying Automatic Emotion Recognition (AER) systems, along 50 ethical considerations. He specifically emphasizes on the risks of privacy violations, reinforcing biases, and potential misuse in surveillance or manipulation. This Ethics Sheet serves as a guide for responsible AER development, and encourages researchers to question *why they automate, whose interests are served, and how success is measured.*

Recognizing the ethical risks and potential misuse of SER technologies, we strongly caution against issues such as biases in emotion datasets, AI models enforcing rigid norms on emotional expression, and the exclusion of neurodiverse and marginalized groups. These concerns must be carefully addressed before deploying SER systems in real-world applications. We urge industries to adopt responsible, explainable, and inclusive AI development practices, ensuring that these technologies are fair, transparent, and beneficial to all users.

Acknowledgments

We extend our gratitude to OrbMedic Inc. (Sonny Chaiwala), Ottawa, Canada, for their financial and technical support throughout this project. Their expertise and insights were pivotal in shaping the direction of our research and ensuring its successful execution. Additionally, we acknowledge the support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Centres of Innovation (OCI).

References

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

Lisa Feldman Barrett, Batja Mesquita, Kevin N Ochsner, and James J Gross. 2007. The experience of emotion. *Annu. Rev. Psychol.*, 58(1):373–403.

A Batliner, S Steidl, and E Nöth. 2006. Releasing a thoroughly annotated and processed spontaneous emotional database: the fau aibo emotion corpus. In *Programme of the Workshop on Corpora for Research on Emotion and Affect*, page 28.

Gulmira Bekmanova, Banu Yergesh, Altynbek Sharipbay, and Assel Mukanova. 2022. Emotional speech recognition method based on word transcription. *Sensors*, 22(5):1937.

Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, and Amena Mahmoud. 2022. Text-based emotion recognition using deep learning approach. *Computational Intelligence and Neuroscience*, 2022(1):2645381.

Dmitri Bitouk, Ragini Verma, and Ani Nenkova. 2010. Class-level spectral features for emotion recognition. *Speech communication*, 52(7-8):613–625.

Danny Boyle. 1996. Trainspotting film. *United Kingdom: Polygram Filmed Entertainment*.

Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, Benjamin Weiss, et al. 2005. A database of german emotional speech. In *Interspeech*, volume 5, pages 1517–1520.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Adil Chakhtouna, Sara Sekkate, et al. 2024. Unveiling embedded features in wav2vec2 and hubert msodels for speech emotion recognition. *Procedia computer science*, 232:2560–2569.

HongYi Chen, Kai Ling Chin, and Chrystalle B.Y. Tan. 2021. Selection and validation of emotional videos: Dataset of professional and amateur videos that elicit basic emotions. *Data in Brief*, 34:106662.

Zebang Cheng, Zhi-Qi Cheng, Jun-Yan He, Kai Wang, Yuxiang Lin, Zheng Lian, Xiaojiang Peng, and Alexander Hauptmann. 2024. Emotion-llama: Multimodal emotion recognition and reasoning with instruction tuning. *Advances in Neural Information Processing Systems*, 37:110805–110853.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Soumya Dutta and Sriram Ganapathy. 2024. Leveraging content and acoustic representations for efficient speech emotion recognition. *arXiv preprint arXiv:2409.05566*.

Paul Ekman. 1992. Facial expressions of emotion: New findings, new questions.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462.

Weiquan Fan, Xiangmin Xu, Xiaofen Xing, Weidong Chen, and Dongyan Huang. 2021. Lssed: a large-scale dataset and benchmark for speech emotion recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 641–645. IEEE.

Crystal A Gabert-Quillen, Ellen E Bartolini, Benjamin T Abravanel, and Charles A Sanislow. 2015. Ratings for emotion film clips. *Behavior research methods*, 47:773–787.

Sayan Ghosh, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2016. Representation learning for speech emotion recognition. In *Interspeech*, pages 3603–3607.

Ziwei Gong, Qingkai Min, and Yue Zhang. 2023. Eliciting rich positive emotions in dialogue generation. In *Proceedings of the First Workshop on Social Influence in Conversations (SICon 2023)*, pages 1–8.

Kenta Hama, Atsushi Otsuka, and Ryo Ishii. 2024. Emotion recognition in conversation with multi-step prompting using large language model. In *International Conference on Human-Computer Interaction*, pages 338–346. Springer.

Dini Handayani, Abdul Wahab, and Hamwira Yaacob. 2015. Recognition of Emotions in Video Clips: The Self-Assessment Manikin Validation. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 13(4):1343.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Elvan Arıkan İyilikci, Merve Boğa, Elif Yüvrük, Yıldız Özkılıç, Osman İyilikci, and Sonia Amado. 2024. An extended emotion-eliciting film clips set (egefilm): assessment of emotion ratings for 104 film clips in a turkish sample. *Behavior Research Methods*, 56(2):529–562.

Philip Jackson and SJUoSG Haq. 2014. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*.

Mimansa Jaiswal and Cristian-Paul Bara. 2020. Muse: a multimodal dataset of stressed emotion. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Tony Kaye. 1998. American history x, new line cinema. *Cited on*, page 99.

Dacher Keltner, Disa Sauter, Jessica Tracy, and Alan Cowen. 2019. Emotional expression: Advances in basic emotion theory. *Journal of nonverbal behavior*, 43:133–160.

Mustaqeem Khan, Wail Gueaieb, Abdulmotaleb El Saddik, and Soonil Kwon. 2024. Mser: Multimodal speech emotion recognition using cross-attention with deep fusion. *Expert Systems with Applications*, 245:122946.

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

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, and Te-Won Lee. 2003. Emotion recognition by speech signals. In *Interspeech*, pages 125–128. Citeseer.

Matthias Ley, Maria Egger, and Sten Hanke. 2019. Evaluating methods for emotion recognition based on facial and vocal features. In *AmI (Workshops/Posters)*, pages 84–93.

Keding Li, Xunbing Shen, Zhencai Chen, Liping He, and Zhennan Liu. 2021. *Effectiveness of Emotion Eliciting of Video Clips: A Self-report Study*, pages 523–542. Springer International Publishing.

MS Likitha, Sri Raksha R Gupta, K Hasitha, and A Upendra Raju. 2017. Speech based human emotion recognition using mfcc. In *2017 international conference on wireless communications, signal processing and networking (WiSPNET)*, pages 2257–2260. IEEE.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Reza Lotfian and Carlos Busso. 2017. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483.

Jiayuan Ma, Hongbin Na, Zimu Wang, Yining Hua, Yue Liu, Wei Wang, and Ling Chen. 2025. Detecting conversational mental manipulation with intent-aware prompting. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9176–9183.

Luz Martinez-Lucas, Mohammed Abdelwahab, and Carlos Busso. 2020. The msp-conversation corpus. *Interspeech 2020*.

Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. 2011. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE transactions on affective computing*, 3(1):5–17.

Tao Meng, Fuchen Zhang, Yuntao Shou, Wei Ai, Nan Yin, and Keqin Li. 2024. Revisiting multimodal emotion recognition in conversation from the perspective of graph spectrum. *arXiv preprint arXiv:2404.17862*.

Yanina Michelini, Ignacio Acuña, Juan Ignacio Guzmán, and Juan Carlos Godoy. 2019. Latemo-e: a film database to elicit discrete emotions and evaluate emotional dimensions in latin-americans. *Trends in Psychology*, 27(2):473–490.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34.

Saif M Mohammad. 2022. Ethics sheet for automatic emotion recognition and sentiment analysis. *Computational Linguistics*, 48(2):239–278.

Agnes Moors. 2009. Theories of emotion causation: A review. *Cognition and emotion*, 23(4):625–662.

Agnes Moors. 2024. An overview of contemporary theories of emotions in psychology. *Emotion Theory: The Routledge Comprehensive Guide: Volume I: History, Contemporary Theories, and Key Elements*.

Louis-Philippe Morency and Tadas Baltrušaitis. 2017. Multimodal machine learning: integrating language, vision and speech. In *Proceedings of the 55th annual meeting of the association for computational linguistics: Tutorial abstracts*, pages 3–5.

Daniel Myrick, Eduardo Sánchez, Heather Donahue, Michael Williams, and Joshua Leonard. 1999. *The Blair witch project*. Artisan Entertainment Santa Monica, CA.

Andrew J. Cohen Nicholas Stoller and Brendan O'Brien. 2014. *Neighbors [Film]*. Universal Pictures.

Minxue Niu, Amrit Romana, Mimansa Jaiswal, Melvin McInnis, and Emily Mower_Provost. 2023. Capturing mismatch between textual and acoustic emotion expressions for mood identification in bipolar disorder. In *Interspeech*. Interspeech.

Bee Suan Ong and Perfecto Herrera. 2005. Semantic segmentation of music audio. In *Proceedings of the International Computer Music Conference*, page 61.

OpenAI. 2023. Models - openai api. `https://platform.openai.com/docs/models/gpt-4o`. Accessed: 2025-01-24.

Alessandro Orso and Tao Xie. 2008. Bert: Behavioral regression testing. In *Proceedings of the 2008 international workshop on dynamic analysis: held in conjunction with the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2008)*, pages 36–42.

Rebecca Passonneau. 2006. Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation. In *5th International Conference on Language Resources and Evaluation, LREC 2006*.

Yushan Qian, Bo Wang, Shangzhao Ma, Wu Bin, Shuo Zhang, Dongming Zhao, Kun Huang, and Yuexian Hou. 2023. Think twice: A human-like two-stage conversational agent for emotional response generation. *arXiv preprint arXiv:2301.04907*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. 2013. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE.

Robert Rodriguez. 2007. *Planet Terror [Film]*. Dimension Films.

Lantana Dioren Rumpa, Adhi Dharma Wibawa, Mauridhi Heri Purnomo, and Harmelia Tulak. 2015. Validating video stimulus for eliciting human emotion: A preliminary study for e-health monitoring system. In *2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, pages 208–213. IEEE.

Alexandre Schaefer, Frédéric Nils, Xavier Sanchez, and Pierre Philippot. 2010. Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers. *Cognition and emotion*, 24(7):1153–1172.

Lone Scherfig. 2011. *One Day [Film]*. Focus Features.

Ewa Siedlecka and Thomas F Denson. 2019. Experimental methods for inducing basic emotions: A qualitative review. *Emotion Review*, 11(1):87–97.

M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic. 2012. A Multimodal Database for Affect Recognition and Implicit Tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55.

Oliver Stone. 1986. *Platoon [Film]*. Orion Pictures.

Kamal Taha. 2024. Text regression analysis: A review, empirical, and experimental insights. *IEEE Access*.

Panagiotis Tzirakis, Anh Nguyen, Stefanos Zafeiriou, and Björn W Schuller. 2021. Speech emotion recognition using semantic information. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6279–6283. IEEE.

Ismail Rasim Ulgen, Zongyang Du, Carlos Busso, and Berrak Sisman. 2024. Revealing emotional clusters in speaker embeddings: A contrastive learning strategy for speech emotion recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12081–12085. IEEE.

Kannan Venkataramanan and Haresh Rengaraj Rajamohan. 2019. Emotion recognition from speech. *arXiv preprint arXiv:1912.10458*.

James Wan. 2013. *The Conjuring [Film]*. Warner Bros. Pictures.

Dong Wang, Lie Lu, and Hong-Jiang Zhang. 2003. Speech segmentation without speech recognition. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Siqing Wu, Tiago H Falk, and Wai-Yip Chan. 2011. Automatic speech emotion recognition using modulation spectral features. *Speech communication*, 53(5):768–785.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. BERxiT: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 91–104, Online. Association for Computational Linguistics.

Xinzhou Xu, Jun Deng, Nicholas Cummins, Zixing Zhang, Li Zhao, and Björn W Schuller. 2021. Exploring zero-shot emotion recognition in speech using semantic-embedding prototypes. *IEEE Transactions on Multimedia*, 24:2752–2765.

Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *2018 IEEE spoken language technology workshop (SLT)*, pages 112–118. IEEE.

YouTube. 2011. Baby laughing hysterically at ripping paper. `https://www.youtube.com/watch?v=RP4abiHdQpc`.

YouTube. 2022a. Mom cat shows baby kittens that golden retriever is safe for them. `https://www.youtube.com/watch?v=N2hssAyomdU`.

YouTube. 2022b. The saddest video ever captured. `https://www.youtube.com/watch?v=Xjwu2Auecko`.

Sara Zhalehpour, Onur Onder, Zahid Akhtar, and Cigdem Eroglu Erdem. 2016. Baum-1: A spontaneous audio-visual face database of affective and mental states. *IEEE Transactions on Affective Computing*, 8(3):300–313.

Huan Zhao, Nianxin Huang, and Haijiao Chen. 2024. Knowledge enhancement for speech emotion recognition via multi-level acoustic feature. *Connection Science*, 36(1):2312103.

Howard Zieff and Laurice Elehwany. 1991. *My Girl [Film]*. Columbia Pictures.

Barbra Zupan and Michelle Eskritt. 2020. Eliciting emotion ratings for a set of film clips: A preliminary archive for research in emotion. *The Journal of Social Psychology*, 160(6):768–789.

Generalizability of Media Frames: Corpus creation and analysis across countries

Agnese Daffara[1], Sourabh Dattawad[1], Sebastian Padó[1], Tanise Ceron[2]
[1]Institute for Natural Language Processing, University of Stuttgart, Germany
[2]Bocconi University, Italy
`{agnese.daffara,pado}@ims.uni-stuttgart.de`
`tanise.ceron@unibocconi.it`

Abstract

Frames capture aspects of an issue that are emphasized in a debate by interlocutors and can help us understand how political language conveys different perspectives and ultimately shapes people's opinions. The Media Frames Corpus (MFC) is the most commonly used framework with categories and detailed guidelines for operationalizing frames. It is, however, focused on a few salient U.S. news issues, making it unclear how well these frames can capture news issues in other cultural contexts. To explore this, we introduce `FrameNews-PT`, a dataset of Brazilian Portuguese news articles covering political and economic news and annotate it within the MFC framework. Through several annotation rounds, we evaluate the extent to which MFC frames generalize to the Brazilian debate issues. We further evaluate how fine-tuned and zero-shot models perform on out-of-domain data. Results show that the 15 MFC frames remain broadly applicable with minor revisions of the guidelines. However, some MFC frames are rarely used, and novel news issues are analyzed using general 'fall-back' frames. We conclude that cross-cultural frame use requires careful consideration.

1 Introduction

Frames are interpretation schemes used to organize reality by isolating and highlighting salient aspects of it (Otmakhova et al., 2024). They combine three levels: cognitive (concerning the mental representations of the world); semantic (concerning the linguistic structures involved); and communicative (their usage and impact on the audience).

Frames are finding increasing application in NLP as a framework to analyze social media and news coverage. In particular the communicative dimension of frames can be seen as a bridge between belief and action (Snow and Benford, 2005). Following Entman (1993), frames have four objectives: i) promote a particular definition of a problem; ii) promote an interpretation of it; iii) propose a moral evaluation; and iv) recommend a treatment or a solution. This makes them a valuable tool for identifying, measuring, and quantifying particular worldviews and communication strategies (Card et al., 2015; Mendelsohn et al., 2021).

Frames are primarily a methodological framework, and not a fixed set of categories. However, a widely used set is based on the Media Frames Corpus (MFC), 15 frames (see Table 1) with annotation guidelines, developed to cover a limited number of issues concerning the U.S. context, e.g., gun control and climate change (Card et al., 2015).

As in other NLP tasks, however, *generalization* is an issue: coarse-grained frames risk failing to capture important aspects of a debate, notably ideological bias, while fine-grained frames may not generalize across debates. For example, Mendelsohn et al. (2021) demonstrate that in news about migration, the generic frame *cultural identity* is correlated with liberal texts; introducing two related sub-frames *hero: cultural diversity* and *threat: national cohesion* leads to a clear separation by political ideology – liberal in the former and conservative the latter case – but these subframes are presumably irrelevant for other debates. Similar questions arise in cross-lingual contexts: the MFC frames were applied by SemEval 2023 Task 3 to 9 languages (Piskorski et al., 2023b), but the applicability of the frames was not assessed, nor was inter-annotator agreement reported.

Indeed, full cross-linguistic generalization of frames is far from clear, given previous work on the cross-lingual generalization of semantic inventories (Peters et al., 1998; Reddy et al., 2017; Gilardi and Baker, 2018). For full generalization, frames would show high *agreement* – their definitions and annotation guidelines would allow each markable in a new language to be assigned to a unique frame – and *completeness* – for each instance in the new language, there would be a suitable frame. However,

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 83–99
November 8-9, 2025 ©2025 Association for Computational Linguistics

we do not have empirical evidence so far.

Thus, we set out to answer these questions:

1. How high are agreement and completeness for MFC frames when annotating a representative sample of news articles despite having been originally developed on a small number of issues?
2. How high are agreement and completeness of the MFC frames when applied to news reporting from other countries?
3. Even if the frames generalize well, is it possible to transfer computational models of frame prediction between languages?

Our study proceeds by creating `FrameNews-PT`, a corpus of 300 news articles of 'hard news' (Reinemann et al., 2012) in Brazilian Portuguese.[1] `FrameNews-PT` is based on the News Portal Recommendation dataset (Lucas et al., 2023). It contains Brazilian news about, e.g., new regulations, taxes, and finance. We manually annotate the articles with MFC frames using a *perspectivist* approach (Cabitza et al., 2023). We carry out several rounds of annotation to a) adapt the MFC guidelines while maintaining their level of detail, b) value the individual perspectives of annotators, and c) examine generalizability and overlap of frames (RQ1+2).

After corpus creation, we conduct modeling experiments (RQ3). We use two transformer-based models fine-tuned on the MFC and 9 chat-instructed models in a zero-shot setting, and analyze how well models perform out of domain on `FrameNews-PT` in comparison with the MFC.

Our results are generally promising: (Most) MFC frames are annotated with high agreement in Brazilian hard news, with little confusion among them. However, some frames appear rarely, being bound to U.S.-specific issues. Also, completeness is not perfect: Brazilian issues outside the scope of the original MFC are often annotated with relatively general frames as a fallback (e.g., *Economic*), leading to some loss of information. On the modeling side, we find that zero-shot models perform better at frame prediction on the Brazilian data than transferred English classifiers, indicating that the data distributions are sufficiently dissimilar. Overall, use of MFC frames in new languages and issues calls for a reflected process.

2 Related Work

Previous work in NLP has investigated how framing political language influences society and public opinion (Lecheler et al., 2015), and how it shapes the spread of information (Gilardi et al., 2021) across communication contexts, such as agenda-setting campaigns (Tsur et al., 2015; Field et al., 2018) and news media (Card et al., 2015). Frames can help analyze argumentation, and frame detection can be seen as argument assessment (Lauscher et al., 2022). Given their connection to political ideologies (Alashri et al., 2015), frames are also useful for analyzing and diversifying perspectives, e.g. in news recommendation (Mulder et al., 2021; Dattawad et al., 2025).

The task of frame detection has often been approached through topic modeling with unsupervised methods (Nguyen et al., 2013; Roberts et al., 2014; Tsur et al., 2015; Ajjour et al., 2019), which, however, tend to be highly corpus-specific. Other studies employ supervised classification techniques, including logistic regression (Card et al., 2016), neural networks (Naderi and Hirst, 2017; Liu et al., 2019), and fine-tuning of pre-trained models (Kwak et al., 2020), all of which require robust gold standard annotations.

The Media Framing Corpus (MFC) (Card et al., 2016), building on the foundational work of Boydstun et al. (2014), introduced a novel annotation framework that has become a reference for frame identification (cf. Section 3). Several studies either use parts of the MFC or applied its guidelines to related datasets, e.g., Kwak et al. (2020) conduct a systematic analysis of the 15 frames in New York Times articles across 17 years.

Khanehzar et al. (2019) are the first to test the generalizability of MFC frames to other contexts. They apply the framework to Australian parliamentary speeches on same-sex marriage and migration, and fine-tuned three BERT-based classifiers. Their findings show a drop in accuracy when the models are applied across different contexts. It is unclear, however, if this is due to the failure of the frames to *generalize*, or only to a shift in frame *distribution*.

Other corpora and labeling schemes have been proposed, though they often focus on specific debate issues. Examples include the Ballistic Missile Defense corpus (Morstatter et al., 2018), the Gun Violence Corpus (Liu et al., 2019), and the migration corpus by Mendelsohn et al. (2021).[2]

---

[1] The corpus is available for the community at this link: `https://github.com/tceron/FrameNews-PT`

[2] For a comprehensive overview of framing detection and

SemEval-2023 Task 3 introduced a sub-task on framing detection in online news within a multilingual setting (Piskorski et al., 2023b). The dataset spans multiple topics and languages. It is annotated with an adaptation of the MFC framework (Piskorski et al., 2023a) and the proposed models, based on Transformer architectures, achieved strong performance. The focus of the study was on modeling, however, not on assessing frame generalizability.

3 Data

The Policy Frames Codebook A key contribution to the study of frames is the Policy Frames Codebook by Boydstun et al. (2014), which introduced 15 framing dimensions applied to three debate issues: immigration, tobacco, and same-sex marriage. The guidelines for frame annotation in this codebook have been dynamically updated over time (Boydstun and Gross, 2014). This work led to the creation of the Media Frames Corpus (Card et al., 2015), later expanded to include three additional issues: gun control, death penalty, and climate (MFC v4.0). The MFC was explicitly developed for three debate issues and the U.S. context, which is a potential limitation. Additionally, as noted by Ali and Hassan (2022), it may not distinguish sufficiently between issues and frames by introducing overly broad categories, such as *Economic*.

For our study, we adopt the 15 framing dimensions as labeled in Card et al. (2015), cf. full list in Table 1. Our guidelines are a version of the Policy Frames Codebook, shortened from 45 to 9 pages to preserve the original information while making it more accessible. We remove specific edge cases and marginal notes, but our annotators are encouraged to consult the original guidelines in cases of doubt. Our version of the guidelines, in English, includes a description for each frame and the cues to recognize it (cf. Appendix A.6). In addition, they include: i) a contextualization of the project, presenting the tool used for annotation (Google Sheets with drop-down menus), and ii) for each frame, an example drawn from MFC headlines on migration where annotators agreed on the frame. In contrast to the version used in SemEval 2023 (Piskorski et al., 2023a), our guidelines retain all relevant content from the original codebook. The guidelines are made available in the repository.

Frame	**Example**
Capacity and Resources	*Immigration debate: Illegals take jobs from Americans*
Crime and Punishment	*Two charged in deaths of illegal immigrants in truck*
Cultural Identity	*Ethnic shift: Immigration—an Irish enclave learns a new language; Mexican immigrants boost a growing Latino population*
Economic	*Society makes no-interest loans to New York's immigrants*
Fairness and Equality	*Strict immigration law unfairly targets Hispanics*
External Regulation and Reputation	*'International village' gets hostile reception*
Health and Safety	*Colombian drug violence leads to exodus*
Legality, Constitutionality and Jurisprudence	*House approves bill to abolish INS; The Senate will begin work next week on its own measure dealing with the immigration agency*
Morality	*County's undocumented workers say they aren't here to 'steal'*
Other	*U.S. under pressure to carry bigger load*
Policy Prescription and Evaluation	*President Donald Trump stalls on promise to eliminate J-1 visa program*
Political	*Following Trump voter fraud allegations, claim that 5.7 million non-citizens voted is wrong*
Public Opinion	*Immigration: Political evangelicals feel push to take sides*
Quality of Life	*Big money, cheap labor*
Security and Defense	*Decision on refugees overdue; U.S. officials must loosen immigration restrictions*

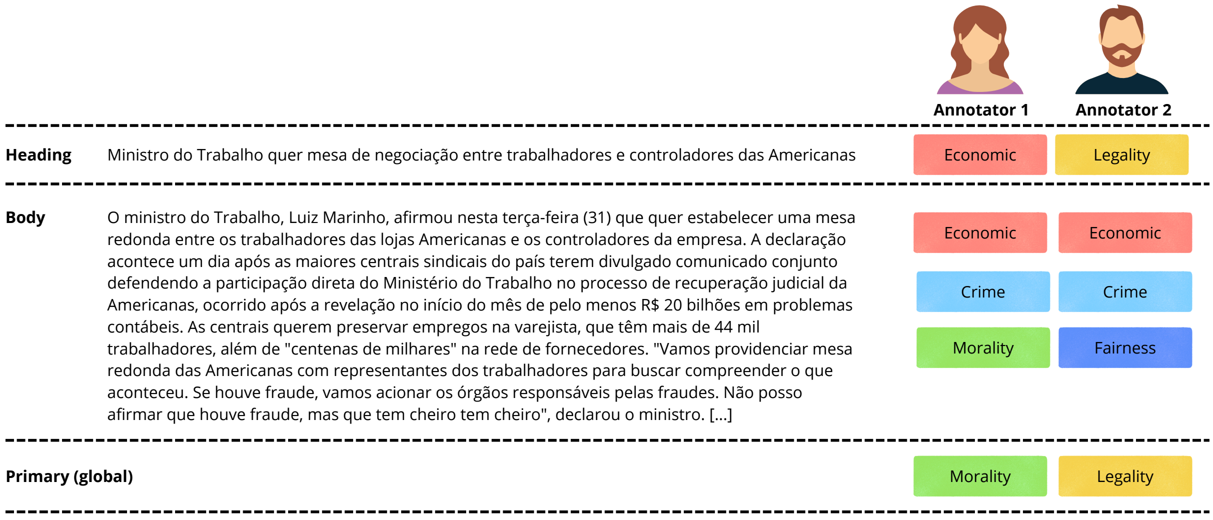
Table 1: The 15 frames and their examples extracted from the MFC headlines on migration.

Data collection Our dataset consists of 300 news articles randomly extracted from the News Portal Recommendation Corpus (NPR), a collection of 148,099 news articles in Portuguese from a Brazilian news portal called G1 with user click history (Lucas et al., 2023). The inclusion of click data makes it suitable for potential future applications in multi-perspective, frame-based news recommendation. The dataset was selected not only for its relevance to recommendation scenarios, but also because it covers a broad range of issues from a cultural and linguistic context different from the U.S. and the Global North generally. This allows us to investigate our research questions on the generalizability of framing annotations and model performance across news issues.

We select articles belonging to the economics and politics sections of the newspaper. The articles

---

analysis, see Ali and Hassan (2022) and Otmakhova et al. (2024), offering a multidisciplinary perspective on the field.



Ministro do Trabalho quer mesa de negociação entre trabalhadores e controladores das Americanas

O ministro do Trabalho, Luiz Marinho, afirmou nesta terça-feira (31) que quer estabelecer uma mesa redonda entre os trabalhadores das lojas Americanas e os controladores da empresa. A declaração acontece um dia após as maiores centrais sindicais do país terem divulgado comunicado conjunto defendendo a participação direta do Ministério do Trabalho no processo de recuperação judicial da Americanas, ocorrido após a revelação no início do mês de pelo menos R$ 20 bilhões em problemas contábeis. As centrais querem preservar empregos na varejista, que têm mais de 44 mil trabalhadores, além de "centenas de milhares" na rede de fornecedores. "Vamos providenciar mesa redonda das Americanas com representantes dos trabalhadores para buscar compreender o que aconteceu. Se houve fraude, vamos acionar os órgãos responsáveis pelas fraudes. Não posso afirmar que houve fraude, mas que tem cheiro tem cheiro", declarou o ministro. [...]

Figure 1: Example of an annotation in `FramesNews-PT` (full article and translation in English are provided in Appendix A.2). Each annotator assigns, in this order: (i) one frame to the article **heading**, (ii) multiple frames to the article **body**, and (iii) a **primary** frame to the article as a whole.

in the NPR dataset often belong to more than one category, so they also span across other news categories as well. In this way, we ensure that articles i) consist of hard news, typically including topics such as foreign and domestic politics, economy and finance (Reinemann et al., 2012), and ii) cover a broad domain, spanning multiple debate issues. A topic modeling analysis with BerTopic (Grootendorst, 2022) on NPR and `FrameNews-PT` shows that our selection of topics is representative of the whole NPR corpus (see Appendix A.3). To comply with model input length limitations, we limit our selection to articles shorter than 300 words and remove boilerplate (banners, ads).

Data annotation We hired two annotators, both Master's students: one Brazilian and one Portuguese. We expect their cultural backgrounds to influence the interpretation and application of frames during the task. In order to acknowledge individual perspectives and subjectivity, we keep the annotations disaggregated and we evaluate the models on each annotator, adopting a perspectivist approach (Cabitza et al., 2023).

Like in the MFC, our annotators label in the following order: (i) the heading frame (single label), (ii) the body frames (multiple labels), and (iii) the primary frame, i.e., the most prominent frame throughout the article (single label). Following the MFC guidelines, the primary frame must be selected from among the annotated frames; in case of doubt, it defaults to the heading frame. Unlike the MFC, however, our annotations are performed at the article level. While span-level annotation may be preferable for certain applications, article-level annotation can still supports various tasks, such as news recommendation with frame diversification. Figure 1 presents an example annotation.

In the MFC, the annotators also decide whether an article is relevant and do not carry out annotations on irrelevant articles[3]. We skip this annotation phase, and therefore annotate all the articles, for two main reasons: i) models are presented with both relevant and potentially irrelevant articles for classification, and we want the annotation process to align with the modeling setup; and ii) our dataset includes only hard news and has already been cleaned. We aggregate irrelevant articles and those that do not match any frame's guidelines under a single label, namely *Other*. This frequently includes news on a personal rather than societal level, such as advice on personal finance, new commercial products, or celebrity news. An example of this category is the following headline, which describes a private decision by President Lula that does not match any frame dimension:

(1) Lula decide não se mudar para a Granja do Torto antes da posse.
Lula decides not to move to Granja do Torto before the inauguration.

---

[3] A relevant article: i) concerns the policy issue in question, ii) takes place in or explicitly regards the United States, iii) has more than four lines, and iv) is a proper article, not a correction of a previous error or an obituary.

The annotation process was organized into four weekly rounds, with an increasing number of articles, based on the assumption that annotators become more efficient and familiar with the task over time: the first two rounds comprised 50 articles each, and the last two rounds contained 100 articles each. After each round, the annotators participated in a discussion session to address ambiguous cases and refined the application of the guidelines. Inter-annotator agreement (IAA) is measured using Krippendorff's α[4] after each round. We hypothesize that i) the IAA increases at each round because of the discussions, and ii) that the score before and after discussion becomes closer at each round.

Annotators' perspectives In the discussion rounds, we observe the influence of the annotators' cultural backgrounds on the usage of frames. The Portuguese annotator struggled to understand certain contexts due to her different cultural perspective and she would sometimes need to look up acronyms, such as STF *(Supremo Tribunal Federal)*. The Brazilian annotator, familiar with the local context, often had a deeper understanding of the reasons and potential consequences of news articles. Consider the following heading sentence, referring to scholarships for students:

(2) Reajuste de bolsas da Capes e do CNPq deve ser anunciado ainda em janeiro, diz Camilo Santana.
Adjustment of Capes and CNPq scholarships should be announced in January, says Camilo Santana.

The Brazilian annotator, aware that access to scholarships is a major issue in Brazil, considered the introduction of such a policy to be significant and labeled the article as *Policy Prescription and Evaluation*. By contrast, the Portuguese annotator, having associated the term *bolsas* with the stock exchange (*bolsa de valores*), labeled it as *Economic*. These differences show that a perspectivist approach is helpful in annotation and modeling. We keep the annotations disaggregated in the evaluation (see Section 5.1 and Table 4 for more details).

4 Analysis of Frame Generalization

We can now investigate the generalization of the MFC frames to the Brazilian context.

Inter-annotator agreement As Figure 2 shows, the agreement score on the primary frame kept improving generally over the four rounds, with the exception of the third round, which contained more complex cases. The discussion was useful in distinguishing difficult frames. After the final revision, the global agreement after discussion on the dataset was α=0.78. It is substantially higher than the one reported on the MFC by (Card et al., 2015), which never exceeds 0.70. This can be attributed to four main reasons: i) in the MFC, new annotators with no previous knowledge were introduced in each round, potentially lowering the overall score; ii) discussing the guidelines with the same person, as we did, could in the worst case lead to overfitting (we alleviate the effects in modeling by keeping annotations disaggregated); iii) annotators usually fell back to general frames rather than specific ones[5]; iv) they used *Other* to annotate articles about personal stories that did not match other frames in the guidelines, consequently raising the agreement.

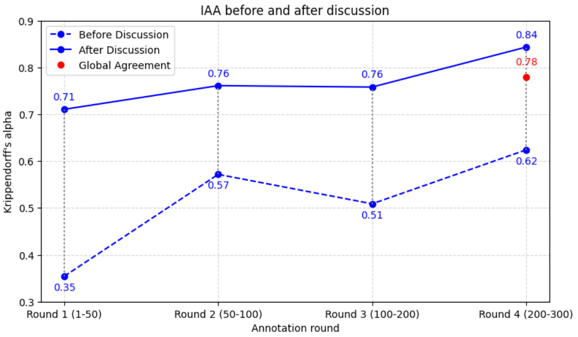


Figure 2: Inter-annotator agreement on the primary frame over four rounds, before and after discussion.

IAA on single frames We also compute agreement on single frames (Figure 3). The results confirm highest agreement for *Other*, arguably due to its particular function in our annotations (see Section 3). Other frames with high scores include *External Regulation and Reputation*, suggesting that it is particularly well defined, *Political*, and *Economic*, which is not surprising given that these are the dominant categories in the dataset. On the other hand, the frames with the lowest agreement are *Capacity and Resources* and *Quality of Life*.

---

[4] Krippendorff's α was adopted following the MFC. It calculates agreement considering the whole pool of labels rather than individual frequency of use (Card et al., 2015).

[5] We use 'general frames' to refer to high-frequency frames — such as *Political*, *Economic*, and *Legality, Constitutionality and Jurisprudence* — that frequently overshadow other frames during annotation. By contrast, 'specific frames' denote low-frequency frames that are more easily obscured by broader categories and potentially tied to specific issues, for example *Capacity and Resources* (see Figures 4, 5, and 6).

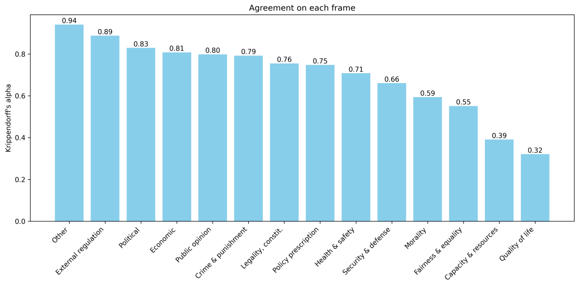


Figure 3: Agreement on each frame (primary).

Frame overlap Figure 4 shows the overlap between the two annotators on primary frames, indicating which frame pairs struggle with high agreement. These include: i) *Crime and Punishment* vs. *Legality, Constitutionality and Jurisprudence* ii) *Fairness and Equality* vs. *Quality of Life*, iii) *Policy Prescription and Evaluation* vs. *Economic*.

Quality of Life and *Fairness and Equality* frequently co-occur when discussing social issues, as in the following:

(3) Ministério diz que novo Bolsa Família terá R$ 18 bilhões para crianças de até 6 anos.
The Ministry says that the new Bolsa Família will have R$ 18 billion for children up to 6 years old.

Crime and Punishment and *Legality, Constitutionality and Jurisprudence* overlap when talking about legal measures to punish someone, while *Policy Prescription* and *Economic* occur together when speaking of policies that affect economy.

We would argue that this overlap is not due to failure of a frame to generalize. Rather, it is a case of a complex situation which is described as having an import on multiple aspects. Such multi-class cases are intrinsic to framing.

Completeness: Frame frequency While a low frequency does not imply that a frame does not generalize, high frequencies for `Other` or general frames may indicate lack of completeness. We therefore compare frame frequencies between MFC and `FrameNews-PT` (Figures 5 and 6).

Some frames, like *Economic*, *Fairness and Equality*, and *External Regulation and Reputation* appeared more frequently in `FrameNews-PT`, suggesting they transfer well to Brazilian debate coverage. In contrast, six frames were used fewer than 20 times: *Health and Safety*, *Security and Defense*, *Quality of Life*, *Capacity and Resources*, *Morality*, and *Public Opinion*. We hypothesize that these frames are not used frequently because they are tied to specific issues in the MFC; for example, *Capacity and Resources* is strongly associated to climate change. *Cultural Identity* was never used by our annotators, which appears due to its specific use in the U.S. issues migration and gun control.

The proportion of *Other* in our corpus is almost the same as in the MFC (13%), although it is used more broadly in our annotations (see Section 3). This means that, even though we sample from a much broader range of issues, the 15 frames still suffice to classify the news. However, annotators in our case tended to fall back to more general frames, indicating a lack of completeness.

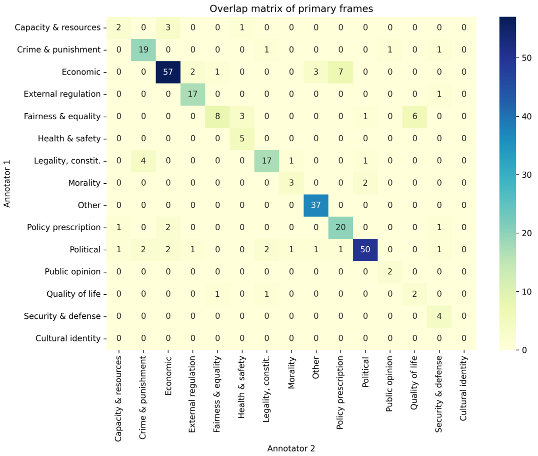


Figure 4: Primary frames overlap between annotators.

Annotator survey To validate our interpretations from above, we asked the annotators to fill a survey about the annotation process, and discussed the results in a meeting. The feedback from the survey confirmed that annotators found our summarized guidelines easier to read and understand than the original one by Boydstun et al. (2014). They identified two main limitations. First, some frame descriptions are overly specific to the U.S. context. For example, *Security and Defense* included immigration issues at the U.S. border (cf. Appendix A.6), a topic not prominent in Brazil. Second, some frames are not described sufficiently distinctly by the guidelines, leading to overlap.

Regarding the frequency of frames, the annotators attributed the low usage of certain frames to the topics covered, confirming our hypothesis that some frames are specific to particular issues. However, when asked whether they would add or eliminate any category, they agreed that the same

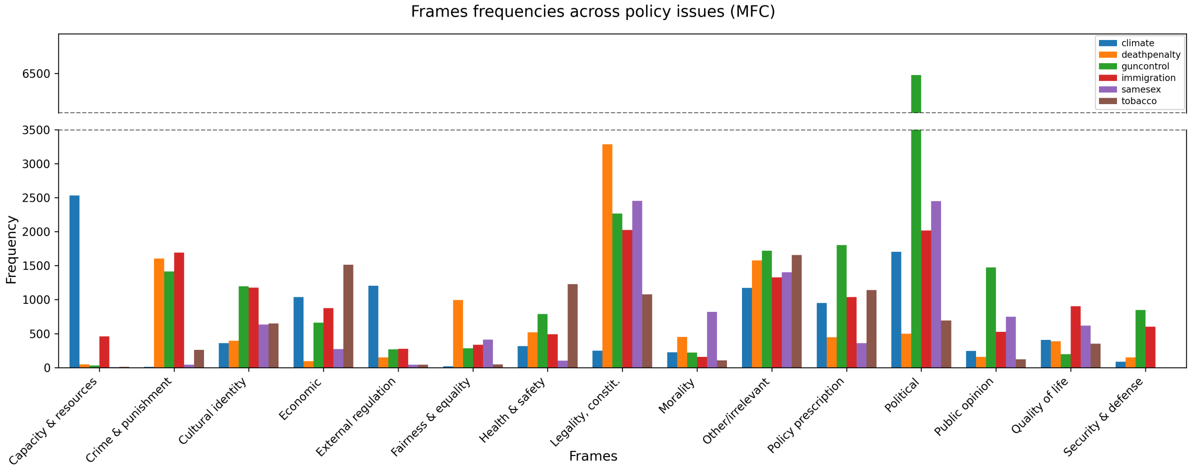


Figure 5: Frames frequencies across policy issues in the MFC.

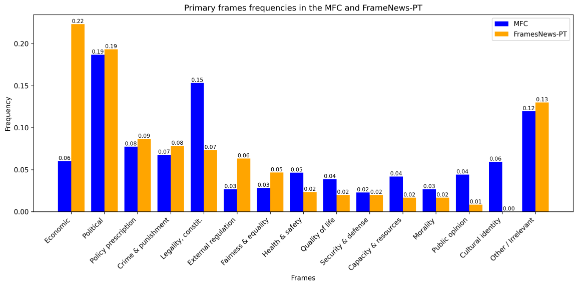


Figure 6: Primary frames frequency in the MFC compared to `FrameNews-PT` (normalized).

tagset could be used for future tasks, provided that some adjustments are made, making the frames more context-aware and contrastive.

The articles tagged as *Other* mainly focused on private issues rather than on societal-level concerns, which is why they did not match any frame. While these could be covered by new frames (e.g., *Personal Finance*, *Advertisement*), adding such fine-grained categories would be misaligned with the goals of framing analysis for hard news.

Conclusions In sum, we find that the 15 frames generalize well enough that they can be employed in future comparable projects, with the following revisions to the guidelines: i) replace U.S.-specific examples with locally relevant counterparts (e.g., substitute *Medicare* with *Bolsa Família*); ii) clarify contrasts between overlapping frames based on the local context (e.g., *Quality of Life* vs. *Health and Safety*); iii) better define the scope of *Other*. The addition of culturally specific frames does not appear necessary, as the existing inventory already provides adequate coverage; rather, adapting the guidelines ensures better fit to the local context.

5 Frame Prediction on `FrameNews-PT`

Even very good frame generalizability does not imply that frame prediction models transfer well across languages, and at least for some frames we see substantial shifts in frequency (cf. Figure 6). Therefore, a) we run two classifiers trained on the MFC corpus, testing them both on MFC and on `FrameNews-PT`; b), we classify `FrameNews-PT` in a zero-shot setting with 8 chat-instructed LLMs ranging from 1B to 12B parameters as well as ChatGPT-4o (exact parameter size unknown). All chosen models are multilingual since `FrameNews-PT` is in Portuguese.

5.1 Experimental Setup

We split the Media Frames Corpus (MFC) into 70% training (21,751 examples), 10% validation (3,107), and 20% test (6,214). The test set is held out and only used for final evaluation. All splits are fixed across runs to ensure consistency. The `FrameNews-PT` corpus is always used as a test set, in order to understand how classifiers perform on a novel set of news issues from a different country.

Supervised classifiers We fine-tune two pre-trained multilingual language models: XLM-RoBERTa-base (Conneau et al., 2019) and Multilingual-E5-base (Wang et al., 2024) for single-label classification over 15 frames. Inputs are tokenized using the model tokenizers with a maximum sequence length of 512, and models are trained using cross-entropy loss. Each model is trained for up to 10 epochs with early stopping based on vali-

Model		Dataset	Accuracy	Global F_1	F_1 (ann 1)	F_1 (ann 2)	Cohen's κ
Majority class baseline		MFC	0.21	-	-	-	-
		`FrameNews-PT`	0.23	-	-	-	-
Fine-tuned	XLM-RoBERTa	MFC	**0.67** ±0.00	**0.68** ±0.00	-	-	-
		`FrameNews-PT`	0.53 ±0.00	0.48 ±0.01	0.49 ±0.00	0.47 ±0.01	-
Fine-tuned	Multilingual-E5	MFC	0.66 ±0.00	0.67 ±0.01	-	-	-
		`FrameNews-PT`	0.56 ±0.01	0.50 ±0.01	0.51 ±0.01	0.49 ±0.01	-
Zero-shot	Qwen2.5-7B-Instruct	MFC	0.40 ±0.10	0.38 ±0.01	-	-	0.50 ±0.00
		`FrameNews-PT`	0.53 ±0.02	0.44 ±0.02	0.45 ±0.03	0.42 ±0.03	0.49 ±0.18
Zero-shot	gpt-4o-2024-08-06	MFC	0.46 ±0.00	0.46 ±0.00	-	-	0.69 ±0.00
		`FrameNews-PT`	**0.59** ±0.01	**0.50** ±0.03	0.51 ±0.04	0.49 ±0.03	0.75 ±0.06

Table 2: Results of the models on the two datasets. For XLM-RoBERTa and Multi-E5, standard deviations are across random seeds. For Qwen and ChatGPT, they are across templates. Accuracy considers the output as correct if it matches at least one annotation. Global F_1 is calculated on the Gold Standard for the MFC, while it is averaged on both annotators for `FramesNews-PT`. Cohen's κ measures reliability across prompt templates. Best results for encoder-based and generative models are **in bold**.

dation loss, using a patience of 3. We use a batch size of 64 for both training and evaluation. The learning rate is 5e-5. All experiments are repeated with five different random seeds.

Zero-shot classifiers For zero-shot classification, we prompt multilingual chat-instructed models from the Qwen (Qwen Team et al., 2024), Google Gemma (Kamath et al., 2025), Llama (Grattafiori et al., 2024), and GPT (Achiam et al., 2023) families. Our prompts include the article to be labeled, the shortened annotation guidelines, and a task description. We evaluate the models on 3 prompt templates (see examples in A.5) and temperature 0.

Evaluation metrics We evaluate the performance of the models on accuracy and F1-score. Accuracy is measured taking the perspectivist approach into account. That is, we consider the answer of the model correct when it matches the primary labels of one of the annotators. The F1-score for the MFC corpus is based on the gold labels while we report the F1-scores for each annotator in `FrameNews-PT` separately. We also calculate Cohen's κ for checking the reliability of the models answers across the evaluated prompt templates.

5.2 Results

We report results for the two supervised models and the two best-performing zero-shot models in Table 2; for the remaining results see the Appendix.

The best model among the supervised learning classifiers for MFC is `RoBERTa-XLM` (acc=67%) while for `FrameNews-PT` it is Multi-E5 (acc=56%). `RoBERTa-XLM` clearly outperforms the zero-shot classifiers on the MFC. On the other hand, ChatGPT-4o is the best model among the zero-shot classifiers for MFC (acc=46%) and `FrameNews-PT` (acc=59%) – not surprisingly, given its parameter count. Note that when comparing ChatGPT-4o and RoBERTa-XLM, ChatGPT-4o yields the best result on `FrameNews-PT`, while RoBERTa-XLM outperforms it by 8 points accuracy on the MFC. The F1-score results follow similar patterns.

These findings suggest a trade-off: supervised models trained on annotated data achieve strong in-domain performance but are less effective in transfer to new corpora, whereas large zero-shot models offer more robust cross-domain performance but fall short in accuracy when compared to supervised models in an in-domain setup.

As for reliability across prompt templates, larger models are more reliable (Ceron et al., 2024): ChatGPT-4o achieves 0.75 and 0.69 Cohen's κ on the MFC and `FrameNews-PT` respectively, whereas Qwen2.5-7B reaches only 0.46 and 0.49.

Figure 7 shows model performance and inter-annotator agreement across frames. The agreement between human annotators is always higher than model-human agreement, except for *Quality of Life*. The models struggle the most with the category *Other* – for the supervised models, this is presumably because this category is very infrequent in the MFC; for the zero-shot models, we hypothesize that this is due to its status as a 'catch-all' category which is hard to learn.

While supervised models perform relatively well across all frames on the MFC (acc=66%-67%), they struggle to generalize to `FrameNews-PT` (acc=53%-

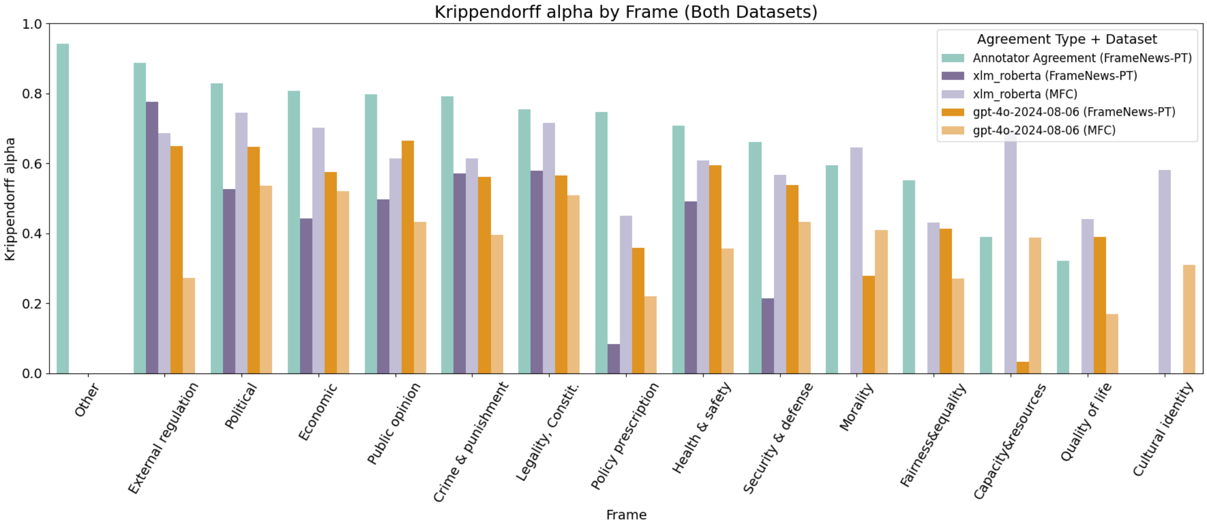


Figure 7: Agreement between models' predictions and annotator 1 on `FrameNews-PT` and MFC, and agreement between annotators on `FrameNews-PT`.

56%), particularly for *Policy Prescription and Evaluation*, *Morality*, *Fairness and Equality*, *Capacity and Resources*, and *Quality of Life*. The latter three also have the lowest inter-annotator agreement. Our best zero-shot model, GPT-4o, performs worst for *Policy Prescription and Evaluation* and *Quality of Life* on the MFC, arguably due to blurred boundaries of these frames (cf. Section 4). On `FrameNews-PT`, it struggles the most with *Morality* and *Capacity and Resources*, likely due to the shift in topics.

These observations give rise to a final question: is the 'limiting factor' in the models' performance on `FrameNews-PT` data quality on the MFC, on `FrameNews-PT`, or the shift in distribution? To understand this, we ran a linear regression to predict GPT-4o's accuracy on the MFC at the frame level (cf. Figure 7). Using three predictors: IAA on MFC, IAA on `FrameNews-PT`, the regression explains 53% of the variance, with only the IAA on the MFC being a significant predictor at $p<0.05$. This is surprising, since GPT-4o is not even trained on the MFC. Our take on this result is that the MFC inventory includes frames with vague or otherwise deficient definitions, which leads to problems both in terms of annotation (low IAA) and in terms of prediction (low performance).

6 Conclusion

This study introduces `FrameNews-PT`, the first dataset of Brazilian Portuguese news with both users' click history (sampled from the NPR corpus) and annotation with the 15 Media Frames Corpus frames. It offers a valuable resource for both the NLP and recommender systems communities. By including frame annotations, it also enables new research directions, such as investigating the relationship between frames and topics.

Our analysis focuses on the generalizability of the MFC frames to the Brazilian context. We find generally high agreement among annotators, which we interpret as a promising sign; however, the MFC frames are not complete. In the absence of specific frames, annotators tend to fall back to general ones, with reduced specificity as consequence. This trend is largely due to the original frames and annotation guidelines being developed for the U.S. context on a limited number of policy issues.

Our modeling experiments show that zero-shot generative models achieve robust results on our dataset. However, the very good in-domain results for supervised MFC models indicate that annotated data is still pivotal to reach strong performance; both model types suffer from shortcomings in frame definition and data creation.

We conclude that the 15 MFC frames remain broadly applicable and continue to capture essential dimensions of political discourse, even across countries and languages. To improve generalization, we recommend refining the annotation guidelines to offer clearer distinctions between overlapping categories and incorporating locally relevant examples (e.g., *Bolsa Família* instead of *Medicare*). Such refinements would enhance the framework's adaptability to diverse cultural contexts and evolving policy debates.

Limitations

The main limitation of our study is the size of our dataset. We had to limit the number of articles to 300, which is a small sample. The main reason is the expense of running this type of study. Annotating frames in news articles is costly because the guidelines are extensive to learn and the articles are long. Annotators also need time to identify all body frames and then the primary frame for each article. For the same reason, we hired only two annotators and chose articles from a single Brazilian news portal, which may limit the generalizability of our results. In modeling, we focused on the primary frame, assuming it represents the article, but considering multiple frames could yield more accurate analysis. Moreover, including few-shot examples could help assess whether they improve performance.

Ethics Statement

The data used in this study is publicly available and was collected from the News Portal Recommendation Corpus (NPR) (Lucas et al., 2023), which contains news articles from Brazilian outlets. The annotation was conducted by two Master's students who were compensated for their work. The study follows ethical guidelines for research involving human participants, ensuring that annotators were informed about the purpose of the study and their rights. The research does not involve sensitive or personal data, and all data is anonymized.

Acknowledgements

We gratefully acknowledge funding from the German Federal Ministry of Research, Technology and Space (BMFTR) under the grant 01IS23072 for the Software Campus project MULTIVIEW.

References

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

Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling frames in argumentation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2922–2932.

Saud Alashri, Sultan Alzahrani, Lenka Bustikova, David Siroky, and Hasan Davulcu. 2015. What animates political debates? analyzing ideological perspectives in online debates between opposing parties. In *Proceedings of the ASE/IEEE International Conference on Social Computing (SocialCom-15)*, Stanford, CA. Academy of Science and Engineering.

Mohammad Ali and Naeemul Hassan. 2022. A survey of computational framing analysis approaches. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9335–9348.

Amber E Boydstun, Dallas Card, Justin H Gross, Philip Resnik, and Noah A Smith. 2014. Tracking the development of media frames within and across policy issues. In *APSA 2014 annual meeting paper*.

Amber E Boydstun and Justin Gross. 2014. Policy frames codebook. *Work in progress*.

Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a perspectivist turn in ground truthing for predictive computing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6860–6868.

Dallas Card, Amber Boydstun, Justin H Gross, Philip Resnik, and Noah A Smith. 2015. The media frames corpus: Annotations of frames across issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444.

Dallas Card, Justin Gross, Amber Boydstun, and Noah A. Smith. 2016. Analyzing framing through the casts of characters in the news. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1410–1420, Austin, Texas. Association for Computational Linguistics.

Tanise Ceron, Neele Falk, Ana Barić, Dmitry Nikolaev, and Sebastian Padó. 2024. Beyond prompt brittleness: Evaluating the reliability and consistency of political worldviews in llms. *Transactions of the Association for Computational Linguistics*, 12:1378–1400.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Sourabh Dattawad, Agnese Daffara, and Tanise Ceron. 2025. Leveraging media frames to improve normative diversity in news recommendations. *arXiv preprint arXiv:2509.02266*.

Robert M Entman. 1993. Framing: Toward clarification of a fractured paradigm. *Journal of communication*, 43(4):51–58.

Anjalie Field, Doron Kliger, Shuly Wintner, Jennifer Pan, Dan Jurafsky, and Yulia Tsvetkov. 2018. Framing and agenda-setting in russian news: a computational analysis of intricate political strategies. *arXiv preprint arXiv:1808.09386*.

Fabrizio Gilardi, Charles R Shipan, and Bruno Wüest. 2021. Policy diffusion: The issue-definition stage. *American Journal of Political Science*, 65(1):21–35.

Luca Gilardi and Collin Baker. 2018. Learning to align across languages: Toward multilingual framenet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Shima Khanehzar, Andrew Turpin, and Gosia Mikolajczak. 2019. Modeling political framing across policy issues and contexts. In *Proceedings of The 17th Annual Workshop of the Australasian Language Technology Association*, pages 61–66.

Haewoon Kwak, Jisun An, and Yong-Yeol Ahn. 2020. A systematic media frame analysis of 1.5 million new york times articles from 2000 to 2017. In *Proceedings of the 12th acm conference on web science*, pages 305–314.

Anne Lauscher, Henning Wachsmuth, Iryna Gurevych, and Goran Glavaš. 2022. Scientia potentia est—on the role of knowledge in computational argumentation. *Transactions of the Association for Computational Linguistics*, 10:1392–1422.

Sophie Lecheler, Mario Keer, Andreas RT Schuck, and Regula Hänggli. 2015. The effects of repetitive news framing on political opinions over time. *Communication Monographs*, 82(3):339–358.

Siyi Liu, Lei Guo, Kate Mays, Margrit Betke, and Derry Tanti Wijaya. 2019. Detecting frames in news headlines and its application to analyzing news framing trends surrounding us gun violence. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 504–514.

Joel Pinho Lucas, João Felipe Guedes da Silva, and Leticia Freire de Figueiredo. 2023. Npr: a news portal recommendations dataset. In *NORMalize@ RecSys*.

Julia Mendelsohn, Ceren Budak, and David Jurgens. 2021. Modeling framing in immigration discourse on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2219–2263, Online. Association for Computational Linguistics.

Fred Morstatter, Liang Wu, Uraz Yavanoglu, Stephen R Corman, and Huan Liu. 2018. Identifying framing bias in online news. *ACM Transactions on Social Computing*, 1(2):1–18.

Mats Mulder, Oana Inel, Jasper Oosterman, and Nava Tintarev. 2021. Operationalizing framing to support multiperspective recommendations of opinion pieces. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 478–488.

Nona Naderi and Graeme Hirst. 2017. Classifying frames at the sentence level in news articles. *Policy*, 9:4–233.

Viet-An Nguyen, Jordan L Ying, and Philip Resnik. 2013. Lexical and hierarchical topic regression. *Advances in neural information processing systems*, 26.

Julia Otmakhova, Shima Khanehzar, and Lea Frermann. 2024. Media framing: A typology and survey of computational approaches across disciplines. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15407–15428.

Wim Peters, Piek Vossen, Pedro Díez-Orzas, and Geert Andriaens. 1998. Cross-linguistic alignment of wordnets with an inter-lingual-index. *Computers and the Humanities*, 32(2):221–251.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, et al. 2023a. News categorization, framing and persuasion techniques: Annotation guidelines. *European Commission, Ispra, JRC132862*.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023b. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

An Yang Qwen Team, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.

Carsten Reinemann, James Stanyer, Sebastian Scherr, and Guido Legnante. 2012. Hard and soft news: A review of concepts, operationalizations and key findings. *Journalism*, 13(2):221–239.

Margaret E Roberts, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G Rand. 2014. Structural topic models for open-ended survey responses. *American journal of political science*, 58(4):1064–1082.

David A Snow and Robert D Benford. 2005. Clarifying the relationship between framing and ideology. *Frames of protest: Social movements and the framing perspective*, 205:209.

Oren Tsur, Dan Calacci, and David Lazer. 2015. A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1629–1638.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

A Appendix

A.1 Frames labels

We adopt the labels from Card et al. (2015). Note that some names from Boydstun et al. (2014) were changed. For example, *Constitutionality and Jurisprudence* was modified to *Legality, Constitutionality and Jurisprudence*, while *Law and Order, Crime and Justice* was changed into *Crime and Punishment*. In the updated Codebook (Boydstun and Gross, 2014), the names of some labels were adapted again (e.g., *Morality* is called *Morality and Ethics*).

A.2 Annotation example

Here we report the complete text of the example news introduced in Figure 1 in Brazilian Portuguese (`PT`) and its translation in English (`EN`).

`PT` **Headline:** Ministro do Trabalho quer mesa de negociação entre trabalhadores e controladores das Americanas.
Body: O ministro do Trabalho, Luiz Marinho, afirmou nesta terça-feira (31) que quer estabelecer uma mesa redonda entre os trabalhadores das lojas Americanas e os controladores da empresa. A declaração acontece um dia após as maiores centrais sindicais do país terem divulgado comunicado conjunto defendendo a participação direta do Ministério do Trabalho no processo de recuperação judicial da Americanas, ocorrido após a revelação no início do mês de pelo menos R$ 20 bilhões em problemas contábeis. As centrais querem preservar empregos na varejista, que têm mais de 44 mil trabalhadores, além de "centenas de milhares" na rede de fornecedores. "Vamos providenciar mesa redonda das Americanas com representantes dos trabalhadores para buscar compreender o que aconteceu. Se houve fraude, vamos acionar os órgãos responsáveis pelas fraudes. Não posso afirmar que houve fraude, mas que tem cheiro tem cheiro", declarou o ministro. Acionistas de referência da Americanas divulgam nota pública A Americanas tem entre os principais acionistas os bilionários Jorge Paulo Lemann, Carlos Sicupira e Marcel Telles. Luiz Marinho questionou nesta terça-feira onde estavam os "compliances" das lojas Americanas, ou seja, as regras de boa gestão e de governança da empresa. "Então é o momento de a gente colocar em dúvida alguns valores do chamado mercado, que é o senhor da razão. Dita regra para lá e para cá, e a regra sagrada não foi observada, que é cuidar da saúde de uma empresa que tem 1.790 lojas espalhadas pelo Brasil e tem 44 mil trabalhadores e trabalhadoras", disse ele. O ministro do Trabalho avaliou, ainda, que a Americanas "pode dar calote no Estado brasileiro, pode dar calote nos credores, nos que estavam de boa fé. E acrescentou que os "espertalhões provavelmente não têm problema". "Quem estava no comando, provavelmente já se livrou", disse. "Os acionistas provavelmente vão poder se locupletar, é possível que tenham se locupletado. Os controladores né. Os acionistas minoritários, coitados, estão na mesma situação dos trabalhadores", concluiu.

`EN` **Headline:** Minister of Labor wants negotiation table between Americanas' workers and controllers.
Body: The Minister of Labor, Luiz Marinho, stated this Tuesday (31) that he wants to establish a roundtable between the workers of the Americanas stores and the company's controllers. The statement comes one day after the country's largest union federations released a joint communiqué defending the direct participation of the Ministry of Labor in Americanas' bankruptcy protection process, which followed the revelation earlier this month of at least R$ 20 billion in accounting irregularities. The unions want to preserve jobs at the retailer, which employs more than 44,000 workers, in addition to "hundreds of thousands" in its supplier network. "We are going to set up a roundtable between Americanas and worker representatives to try to understand what happened. If there was fraud, we will trigger the authorities responsible for fraud cases. I cannot say there was fraud, but it certainly smells like it," said the minister. Major shareholders of Americanas release public note Americanas' main shareholders include billionaires Jorge Paulo Lemann, Carlos Sicupira, and Marcel Telles. On Tuesday, Luiz Marinho questioned where the "compliances" of Americanas were, meaning the company's good management and governance practices. "So this is the moment for us to question certain values of the so-called market, which claims to be the voice of reason. It sets rules here and there, yet the sacred rule was not observed: taking care of the health of a company with 1,790 stores across Brazil and 44,000 workers," he said. The Minister of Labor also argued that Americanas "may default on the Brazilian state, may default on creditors, on those who acted in good faith. And he added that the 'smart operators probably have no problems.' 'Those in charge probably already got away,' he said. "The shareholders will probably be able to enrich themselves, it is possible that they already have. The controllers, right. The minority shareholders, poor them, are in the same situation as the workers," he concluded.

A.3 `FrameNews-PT` and NPR analysis

For the NPR dataset, the average number of words in article titles is 14.9 ± 3.6, while the average number of words in article bodies is 471.7 ± 391.9 (Lucas et al., 2023). Table 3 shows the most frequent topics in NPR and `FrameNews-PT`.

A.4 Expenses

We spent 96 dollars for evaluating ChatGPT-4o with the OpenAI API (3 prompt templates x 6,214

id	`FrameNews-PT`	NPR
0	presidente, lula, bolsonaro, governo, ex, feira, nesta, petrobras, conselho, pt	lula, presidente, governo, ex, petrobras, conselho, feira, ministro, cargo, nesta
1	governo, haddad, sobre, fiscal, tributária, imposto, vai, ministro, câmara, texto	bolsonaro, presidente, federal, atos, stf, tribunal, ex, ministro, república, justiça
2	federal, atos, stf, tribunal, prisão, justiça, torres, ministro, polícia, pf	governo, haddad, fiscal, sobre, tributária, proposta, câmara, regra, ministro, imposto
3	mínimo, bilhões, inflação, salário, 2023, pib, governo, valor, anno, tesouro	juros, taxa, brasil, tesouro, dívidas, bancos, milhões, bilhões, inss, estados
4	dívidas, bancos, tesouro, milhões, bilhões, empréstimos, governo, juros, estados, taxa	café, rural, vídeos, bacalhau, globo, produtos, soja, onde, vem, produção
5	empresa, petrobras, bahia, companhia, vendas, marca, varejo, ford, refinaria, unilever	empresa, petrobras, bahia, companhia, vendas, marca, ford, varejo, refinaria, bilhões
6	rural, bacalhau, globo, vídeos, soja, produtos, saiba, água, assistidos, café	mínimo, inflação, salário, bilhões, 2023, pib, governo, anno, reajuste, família
7	brasil, auxílio, divórcios, mulheres, milhões, número, julho, ativos, datafolha, analisa	microsoft, internet, telemarketing, explorer, uso, bloqueadores, partir, clienti, codice, applicazione
8	microsoft, telemarketing, internet, explorer, ataque, clientes, secretaria, código, aplicativo, partir	crédito, cartão, americanas, causas, trabalho, bancos, linha, trabalhadores, diagnóstico, câmbio

Table 3: Comparison between top topics in `FramesNews-PT` and NPR, detected with BERTopic (Grootendorst, 2022). Topics: ■ politics, ■ economics, ■ justice, ■ production, ■ companies, ■ technology, ■ social support.

= 18,642 prompts). We paid 445 euros for each annotator to annotate the 300 examples in 27 hours of work.

A.5 Prompt templates

```
### PROMPT:
"{content}"

### TASK:
Classify the PROMPT above into exactly ONE of
    these frame categories:

”1” Economic
”2” Capacity and resources
”3” Morality
”4” Fairness and equality
”5” Legality, Constitutionality, Jurisdiction
”6” Crime and punishment
”7” Security and defense
”8” Health and safety
”9” Quality of life
”10” Cultural identity
”11” Public opinion
”12” Political
”13” Policy prescription and evaluation
”14” External regulation and reputation
”15” Other

### CLASSIFICATION GUIDELINES:

{guidelines}

Base your answer only on the PROMPT and the
    guidelines provided above. Answer as a
    single number (”1”, ..., ”15”) corresponding
     to the most appropriate category.

### ANSWER:
```

Listing 1: Prompt template without guidelines: zero1

```
### CLASSIFICATION GUIDELINES:

{guidelines}

### PROMPT:
"{content}"

### TASK:
Classify the PROMPT above into exactly ONE of
    the categories below.
”1” Economic
”2” Capacity and resources
”3” Morality
”4” Fairness and equality
”5” Legality, Constitutionality, Jurisdiction
”6” Crime and punishment
”7” Security and defense
”8” Health and safety
”9” Quality of life
”10” Cultural identity
”11” Public opinion
”12” Political
”13” Policy prescription and evaluation
”14” External regulation and reputation
”15” Other

 Answer as a single number (”1”,..., ”15”)
     corresponding to the most appropriate
     category.
 ### ANSWER:
```

Listing 2: Prompt template with guidelines: The guidelines can be found in the github link.

A.6 Description of frames

Here we provide the descriptions of frames in our guidelines. The full guidelines are provided in the repository.

1. Economic: The costs, benefits, or any monetary/financial implications of the issue (to an individual, family, organization, community or to the economy as a whole). Can include the effect of policy issues on trade, markets, wages, employment or unemployment, viability of specific industries or businesses, implications of taxes or tax breaks, financial incentives, etc.

2. Capacity and resources: The lack or availability of resources (time, physical, geographical, space, human, and financial resources). The capacity of existing systems and resources to carry out policy goals. The easiest way to think about it is in terms of there being "not enough" or “enough” of something. The capacity or resources may be an impediment to solving a problem or adequately addressing an issue.

3. Morality: Any perspective that is compelled by religious doctrine or interpretation, duty, honor, righteousness or any other sense of ethics or social or personal responsibility. It is sometimes presented from a religious perspective (i.e. “eye for an eye”), but non-religious frames can also be used. For example, the moral imperatives to help others can be used to justify military intervention or foreign aid, social programs such as Medicare, welfare, and food stamps. Appeals that a policy move “is just the right thing to do” or “would indicate a recognition of our shared humanity” may reflect humanist morality. The commitment aspect of marriage would evoke feelings of morality. Environmental arguments that focus on responsible stewardship or “leaving something for our children” are based in a sense of responsibility or morality. Lawbreakers, including illegal immigrants, can be presented as fundamentally immoral, conversely breaking a law that is bad or unjust can be presented as moral (e.g., Rosa Parks). Enacting protective legislation, such as laws that protect children from pedophiles, guns, violence, poverty, or failure to do so can also be presented using moral frames.

4. Fairness and equality: The fairness, equality or inequality with which laws, punishment, rewards, and resources are applied or distributed among individuals or groups. Also the balance between the rights or interests of one individual or group compared to another individual or group. Fairness and Equality frame signals often focus on whether society and its laws are equally distributed and enforced across regions, race, gender, economic class, etc. Many gender and race issues, in particular, include equal pay, access to resources such as education, healthcare and housing. Another example could be fairness considerations about whether punishments are proportional to crimes committed. The frame is also used when discussing social justice, discrimination and talk of an inmate’s innocence or exogeneration.

5. Legality, Constitutionality, Jurisdiction: The legal, constitutional, or jurisdictional aspects of an issue. Legal aspects include existinglaws, reasoning on fundamental rights and court cases; constitutional aspects include all discussion of constitutional interpretation and/or potential revisions; jurisdiction includes any discussion of which government body should be in charge of a policy decision and/or the appropriate scope of a body’s policy reach. This frame deals specifically with the authority of government to regulate, and the authority of individuals/corporations to act independently of government. Of special note are constraints imposed on freedoms granted to individuals, government, and corporations via the Constitution, Bill of Rights and other amendments. Some frequent arguments and issues are: i) the right to bear arms; ii) equal protection; iii) free speech and expression; iv) the constitutionality of restricting individual freedoms and imposing taxes; v) conflicts between state, local or federal regulation and authority, or between different branches of government; vi) legal documentation (green card, visas, passports, driver licenses, marriage license, etc.).

6. Crime and punishment: The violation of policies and its consequences. It includes enforcement and interpretation of civil and criminal laws, sentencing and punishment with retribution or sanctions. This frame includes: i) deportation when an individual does not have the necessary documents that grant legal standing; ii) increases or reductions in crime; iii) punishment and execution; iv) resources analysis like DNA analysis. Usually found together with other frames, such as Economic, Legality, constitutionality and jurisdiction, Morality, and Capacity and resources. The primary frame should be chosen according to where the emphasis is.

7. Security and defense: Any threat to a person, group, or nation, or any defense that needs to be taken to avoid that threat. Security and Defense frames differ from Health and Safety frames in that Security and Defense frames address a preemptive action to stop a threat from occurring, whereas Health and Safety frames address steps taken to ensure safety in the event that something happens. It can include efforts to build a border fence or “secure the borders,” issues of national security including resource security, efforts of individuals to secure homes, neighborhoods or schools, and efforts such as guards and metal detectors that would defend children from a possible threat. Discussion regarding terrorist activity should be coded as Security and Defense (e.g. arrests of terrorists, immigrants linked to terrorism activity, increased border security to prevent terrorism). Arrests at the border will receive both a Crime and Punishment and Security and Defense frame but the primary frame would be Security and Defense since the action is taking place on the border. All terrorist attacks are coded as Security and Defense, but attention should be paid to potential criminal, legal, or any other aspects and double coded accordingly.

8. Health and safety: The potential health and safety outcomes of any policy issue (e.g. health care access and effectiveness, illness, disease, sanitation, carnage, obesity, mental health infrastructure and building safety). Also policies taken to ensure safety in case of a tragedy would fit under this (e.g. emergency preparedness kits, lock down training in schools, disaster awareness classes for teachers). It includes any discussion of the various capital punishment methods and procedures and any mentions of refugees. Often used in conjunction with Quality of Life.

9. Quality of life: The benefits and costs of any policy on quality of life. The effects of a policy on people’s wealth, mobility, access to resources, happiness, social structures, ease of day-to-day routines, quality of community life, etc. It includes any mention of people receiving generic “benefits”, adoptions, and weddings. Often used in conjunction with Health and Safety.

10. Cultural identity: The social norms, trends, values and customs constituting any culture(s). It includes: i) language issues and language learning; ii) patriotism and national traditions, the history of an issue or the significance of an issue within a group or subculture; iii) census and demographics; iv) cultural shifts in a group or society; v) cultural norms of ethnic and political groups. May also include stereotypes or assumed preferences and reactions of a group (e.g., an affinity for Republicans to wear cowboy hats); vi) references and quotations of famous people like politicians, leaders or representatives of a subculture.

11. Public opinion: The opinion of the general public. It includes references to general social attitudes, protests, polling and demographic information, as well as any public passage of a proposition or law (i.e. “California voters passed Prop 8”). All the opinions that represent the sentiment of a group will be coded as Public opinion. However, a group of experts in a particular domain gets coded according to their domain (e.g. police officers in Crime and Punishment, or climate scientists in Capacity and Resources).

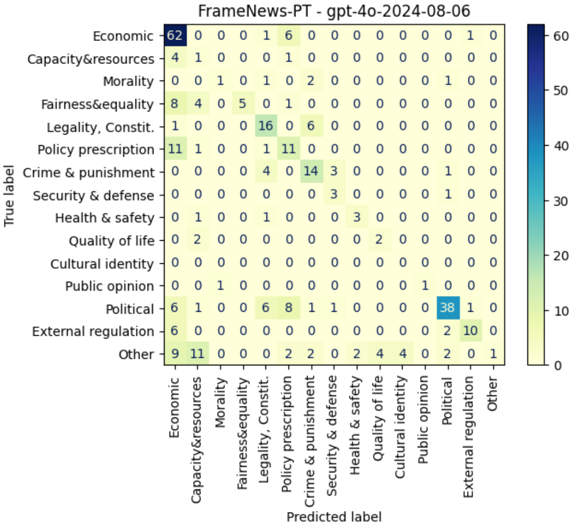
12. Political: In general, any political considerations surrounding an issue. It includes political actions, maneuvering, efforts or stances towards an issue (e.g. partisan filibusters, lobbyist involvement, deal-making and vote trading), mentions of political entities or parties (e.g., Democrats, Republicans, Libertarians, Green Party). When a headline mentions “both sides” this refers to politics.

13. Policy prescription and evaluation: The analysis of whether hypothetical policies will work or existing policies are effective. What is/isn’t currently allowed and what should/shouldn’t be done? “Policy” encompasses formal government regulation (e.g., federal or state laws) as well as regulation by businesses (e.g., sports arenas not allowing the sale of alcohol). This frame dimension—perhaps more than any other—is likely to appear frequently across texts. Yet care should be given to only use this code category as the primary frame when the main thrust of an article is really about policy, for example when it describes the success and failure of existing policies or proposes policy solutions to a problem.

14. External regulation and reputation: In general, the country’s external relations with another nation; the external relations of a state with another.This frame includes: i) trade agreements and outcomes; ii) comparisons of policy outcomes between different groups or regions; iii) the perception of one nation, state, and/or group byanother (for example, international criticisms of the United States maintaining capital punishment); iv) border relations, interstate or international efforts to achieve policy goals; v) alliances or disputes between groups.

15. Other: Any frame signal that does not fit in the first 14 dimensions.

A.7 Further results



	Economic	Capacity&resources	Morality	Fairness&equality	Legality, Constit.	Policy prescription	Crime & punishment	Security & defense	Health & safety	Quality of life	Cultural identity	Public opinion	Political	External regulation	Other
Economic	62	0	0	0	1	6	0	0	0	0	0	0	0	1	0
Capacity&resources	4	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Morality	0	0	1	0	1	0	2	0	0	0	0	0	1	0	0
Fairness&equality	8	4	0	5	0	1	0	0	0	0	0	0	0	0	0
Legality, Constit.	1	0	0	0	16	0	6	0	0	0	0	0	0	0	0
Policy prescription	11	1	0	0	1	11	0	0	0	0	0	0	0	0	0
Crime & punishment	0	0	0	0	4	0	14	3	0	0	0	0	1	0	0
Security & defense	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0
Health & safety	0	1	0	0	1	0	0	0	3	0	0	0	0	0	0
Quality of life	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
Cultural identity	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Public opinion	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
Political	6	1	0	0	6	8	1	1	0	0	0	0	38	1	0
External regulation	6	0	0	0	0	0	0	0	0	0	0	0	2	10	0
Other	9	11	0	0	0	2	2	0	2	4	4	0	2	0	1

Figure 8: Confusion matrix of the best prompt and best model (chatGPT-4o) in comparison with annotator 1.

Model	**Global F_1**	**F_1 (ann 1)**	**F_1 (ann 2)**	**Accuracy**
gpt-4o-2024-08-06	**0.50** ± 0.04	0.51 ± 0.04	0.49 ± 0.03	**0.59** ± 0.02
Qwen2.5-7B-Instruct	0.45 ± 0.04	0.46 ± 0.04	0.43 ± 0.03	0.54 ± 0.02
Llama-3.1-8B-Instruct	0.36 ± 0.04	0.37 ± 0.03	0.34 ± 0.04	0.44 ± 0.04
gemma-3-4b-it	0.35 ± 0.03	0.36 ± 0.02	0.34 ± 0.04	0.43 ± 0.02
Qwen2.5-1.5B-Instruct	0.29 ± 0.09	0.30 ± 0.09	0.28 ± 0.08	0.41 ± 0.08
Qwen2.5-3B-Instruct	0.33 ± 0.08	0.34 ± 0.08	0.32 ± 0.07	0.40 ± 0.07
Llama-3.2-3B-Instruct	0.11 ± 0.07	0.12 ± 0.06	0.10 ± 0.07	0.26 ± 0.06
Llama-3.2-1B-Instruct	0.16 ± 0.02	0.17 ± 0.02	0.15 ± 0.02	0.24 ± 0.03

(a) `FrameNews-PT` zero-shot results. We report F_1 scores averaged on both annotators (global F_1), against annotator 1 (ann1) and annotator 2 (ann 2).

Model	**Global F_1**	**Accuracy**
gpt-4o-2024-08-06	**0.47** ± 0.01	**0.46** ± 0.01
Qwen2.5-7B-Instruct	0.39 ± 0.02	0.40 ± 0.01
Llama-3.1-8B-Instruct	0.35 ± 0.02	0.37 ± 0.02
gemma-3-4b-it	0.34 ± 0.01	0.37 ± 0.03
Qwen2.5-3B-Instruct	0.30 ± 0.05	0.30 ± 0.04
Llama-3.2-3B-Instruct	0.16 ± 0.12	0.27 ± 0.09
Qwen2.5-1.5B-Instruct	0.21 ± 0.12	0.23 ± 0.11
Llama-3.2-1B-Instruct	0.06 ± 0.06	0.12 ± 0.07

(b) `MFC` zero-shot results. We report scores against the gold label.

Table 4: Zero-shot results with generative models on both datasets. Accuracy is calculated by considering the output as correct if it matches at least one annotation. The standard deviation is calculated based on the three prompt templates used to evaluate each model. Best results are **in bold**.

Cross-Lingual Extractive Question Answering with Unanswerable Questions

Yuval Gorodissky[1,2], **Elior Sulem**[1,2], **Dan Roth**[3,4]
[1] Faculty of Computer and Information Science, Institute for Applied AI Research
[2] Data Science Research Center
Ben-Gurion University of the Negev
[3] Department of Computer and Information Science, University of Pennsylvania
[4] Oracle AI
`yuvalgor@post.bgu.ac.il, eliorsu@bgu.ac.il, danroth@seas.upenn.edu`

Abstract

Cross-lingual Extractive Question Answering (EQA) extends standard EQA by requiring models to find answers in passages written in languages different from the questions. The Generalized Cross-Lingual Transfer (G-XLT) task evaluates models' zero-shot ability to transfer question answering capabilities across languages using only English training data. While previous research has primarily focused on scenarios where answers are always present, real-world applications often encounter situations where no answer exists within the given context. This paper introduces an enhanced G-XLT task definition that explicitly handles unanswerable questions, bridging a critical gap in current research. To address this challenge, we present two new datasets: miXQuAD and MLQA-IDK, which address both answerable and unanswerable questions and respectively cover 12 and 7 language pairs. Our study evaluates state-of-the-art large language models using fine-tuning, parameter-efficient techniques, and in-context learning approaches, revealing interesting trade-offs between a smaller fine-tuned model's performance on answerable questions versus a larger in-context learning model's capability on unanswerable questions. We also examine language similarity patterns based on model performance, finding alignments with known language families.[1]

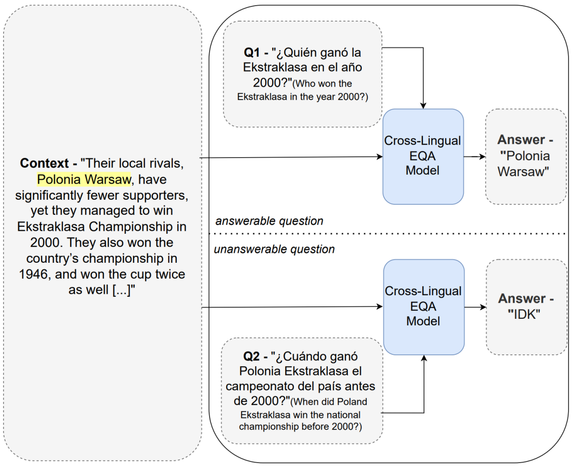

Context - "Their local rivals, Polonia Warsaw, have significantly fewer supporters, yet they managed to win Ekstraklasa Championship in 2000. They also won the country's championship in 1946, and won the cup twice as well [...]"
Q1 - "¿Quién ganó la Ekstraklasa en el año 2000?"(Who won the Ekstraklasa in the year 2000?)
Cross-Lingual EQA Model
Answer - "Polonia Warsaw"
answerable question
unanswerable question
Cross-Lingual EQA Model
Answer - "IDK"
Q2 - "¿Cuándo ganó Polonia Ekstraklasa el campeonato del país antes de 2000?"(When did Poland Ekstraklasa win the national championship before 2000?)

Figure 1: Illustration of Cross-lingual EQA task using MiXQuAD examples. In these examples, questions are in Spanish and contexts/answers are in English, demonstrating both answerable and unanswerable cases. The English translation of each question is provided for reference.

1 Introduction

Extractive Question Answering (EQA) is the task of finding text spans within given contexts that answer given natural language questions. This field was formalized with the Stanford Question Answering Dataset (SQuAD, Rajpurkar et al., 2016), which set a key benchmark for EQA. Recent advances in large language models (LLMs, Brown et al., 2020) have significantly improved EQA performance, marking important progress in Natural Language Understanding. Cross-lingual EQA is a task where the question and its corresponding context are presented in different languages[2], addressing a critical need in today's globalized world. For instance, an English-speaking user might need to query content available only in Chinese or Arabic, challenging the monolingual assumptions in traditional QA systems. Recent research has demonstrated that EQA techniques can be effectively applied to downstream tasks, showing promising results for example in zero-shot event extraction (Lyu et al., 2021) and summarization evaluation (Deutsch et al., 2021; Durmus et al., 2020). While these studies focused on English, their QA-based approaches could naturally extend to cross-lingual scenarios such as cross-lingual summarization (Wang et al., 2022) and cross-lingual event extraction (Subburathinam et al., 2019). Lewis et al. (2020) introduced the Generalized Cross-Lingual Transfer (G-XLT) task, addressing cross-lingual QA in the case where models are trained on English and evaluated on multiple language pairs, assuming that all questions are answerable. However, the ability to identify when a question cannot be answered is crucial for real-world applications.

---

[1] The code and datasets are publicly available at `https://github.com/NLU-BGU/Cross-Lingual-Extractive-Question-Answering-with-Unanswerable-Questions`.

[2] This term is sometimes also used for referring to the case where the training and test corpora are in different languages, while in each of them the question and the context are in the same language (Artetxe et al., 2020).

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 100–121
November 8-9, 2025 ©2025 Association for Computational Linguistics

In this paper, we expand the G-XLT framework by introducing unanswerable questions, as illustrated in Figure 1. This extension reflects real-world scenarios where answers may not exist within the given context. Our approach also broadens the evaluation of Lewis et al. (2020), limited to encoder-based models, to include state-of-the-art models, encompassing large language models (LLMs) and various transformer architectures, providing a more comprehensive assessment of cross-lingual QA capabilities. To address the limitations in current datasets and to provide a robust benchmark for this extended task, we have developed two test datasets: miXQuAD and MLQA-IDK. The miXQuAD dataset combines elements from SQuAD v2.0 and XQuAD (Rajpurkar et al., 2018; Artetxe et al., 2020), integrating unanswerable questions into a multilingual framework. It covers 12 languages and includes a total of 2,072 examples per language, balancing answerable and unanswerable questions. The MLQA-IDK dataset, derived from MLQA (Lewis et al., 2020), specifically focuses on cross-lingual unanswerability, covering 7 language pairs and employing techniques such as antonym and entity augmentation to generate unanswerable questions.

Our cross-lingual QA analysis reveals a trade-off between fine-tuned small models and large models with in-context learning: mT5-large (1.2B parameters) with fine-tuning excelled at detecting unanswerable questions, while AYA-101 (13B parameters) with hint prompting, where the unanswerability option is mentioned in the prompt, performed better on answerable questions. Hint prompting significantly improved unanswerable question detection across models while maintaining performance on answerable questions. Fine-tuned AYA-101 achieved the best performance, outperforming both its regular prompt version and mT5-large across both answerable and unanswerable questions. Out-of-domain testing on MLQA-IDK and open-domain evaluation on XTREME-UP (Ruder et al., 2023) demonstrate model robustness across diverse QA scenarios, including low-resource languages.

We also examine the dependence of the results on the specific languages and on linguistic relationships. First, comparing between cases where the question is in English (English-Questions) to those where the context is in English (English-Contexts), we observe that models performed better when contexts are in English, indicating that processing questions in various languages while keeping English contexts is more manageable. Second, language clustering analysis revealed three groups that align to some extent with language typology—suggesting linguistic relationships influence model behavior.

In an advanced analysis, we examine answerability-related error patterns, test the models' reliance on parametric knowledge, and explore their uncertainty in the different types of prediction. In particular, we observe that hint prompting reduces uncertainty when classifying unanswerable questions, while fine-tuning improves overall certainty but reduces the confidence gap between correct and incorrect predictions.

Our main contributions are the following. First, we expand the Generalized Cross-Lingual Transfer (G-XLT) task to explicitly handle unanswerable questions. Second, we introduce two novel test sets, miXQuAD and MLQA-IDK for the extended task. Third, through the analysis of state-of-the-art models with varying architectures and parameter sizes, we provide insights into performance patterns and language dependency, and reveal trade-offs between model size and training approaches, advancing cross-lingual QA understanding.

2 Related Work

2.1 Extractive Question Answering

EQA is a fundamental Natural Language Understanding (NLU) task that involves identifying and extracting answer spans from a given context in response to natural language questions. This task serves as a critical benchmark in evaluating machine reading comprehension capabilities (Wang et al., 2018). Initially, EQA research focused primarily on monolingual settings, with SQuAD (Rajpurkar et al., 2016) establishing foundational benchmarks through English Wikipedia-derived question-answer pairs. The introduction of BERT (Devlin et al., 2019) marked a significant advancement through its use of bidirectional transformers, though early development remained largely English-centric, with other language datasets often being SQuAD translations such as Arabic (Mozannar et al., 2019) and Spanish (Carrino et al., 2020).

2.2 Evolution of Cross-lingual QA

The development of QA in multiple languages has followed two main strategies: fine-tuning existing models for new languages and developing zero-shot transfer capabilities across languages. Datasets like XQuAD (Artetxe et al., 2020), TyDiQA (Clark et al., 2020), and XTREME (Hu et al., 2020) have facilitated this research through multilingual question-answer pairs. The Multilingual Transfer (XLT) task, introduced with MLQA (Lewis et al., 2020), pioneered generalization from English-trained models to other languages, where questions and contexts are in the same language. Its extension, the G-XLT task, formalized cross-lingual QA by requiring models trained solely on English data to handle questions and contexts in different languages. Recent work has explored retrieval-augmented approaches to address cross-lingual challenges. Cross-lingual QA has also been studied in open-domain QA, where document retrieval is required before answering the question. In particular, XOR-TyDi QA (Asai et al., 2020) revealed severe performance drops when answers exist only in foreign-language documents, requiring retrieval across massive multilingual corpora. More recently, Ranaldi et al. (2025) demonstrated that multilingual RAG systems face unique difficulties when retrieved documents span multiple languages, with performance degrading when models must integrate information across linguistic boundaries. However, all these frameworks assumed questions were answerable. We extend the G-XLT framework to address this gap by introducing two complementary benchmarks: miXQuAD and MLQA-IDK, providing evaluation capabilities across multiple languages with explicit no-answer detection. We also explore the adaptability of our study to open-domain settings in Section 5.4.

2.3 No Answer Importance

The ability to identify unanswerable questions is critical in real-world applications, with NQ (Kwiatkowski et al., 2019) showing that 51% of real queries lack answers in their given context. While this challenge has been studied in English monolingual settings, beginning with SQuAD v2.0 (Rajpurkar et al., 2018), research has shown that even advanced language models struggle with this task, often hallucinating plausible but incorrect answers (Slobodkin et al., 2023). English-focused datasets like HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022) further demonstrated this challenge through multi-document reasoning requirements, with MuSiQue introducing contrast questions to increase evaluation rigor. However, these existing datasets and research have focused on monolingual settings, leaving a gap in understanding how models perform on unanswerable questions in cross-lingual scenarios, which we address in this paper.

3 Task, Dataset Creation And Structure

3.1 The Task

The task of *Generalized Cross-Lingual EQA with IDK* is defined as follows. Given a training dataset $D = \{(c_i, q_i, a_i)\}_{i=1}^{N}$, $N \in \mathbb{N}$, where c_i is a context, q_i is a question, a_i is an answer, and all elements c_i, q_i, a_i are in English, we aim to learn a mapping:

$$f : (q \in L_q, c \in L_c) \rightarrow \begin{cases} s \subseteq c & \text{if answer exists} \\ \text{IDK} & \text{otherwise} \end{cases}$$

where L_c and L_q are the context and question languages respectively, and $L_c = L_q =$ English.

During evaluation, we consider two settings: (i) English-Questions: where questions are in English and contexts are in other languages ($L_q =$ English, $L_c \neq$ English), and (ii) English-Contexts: where contexts are in English and questions are in other languages ($L_c =$ English, $L_q \neq$ English).

3.2 Dataset Creation and Structure

To evaluate cross-lingual EQA capabilities, we present two evaluation test sets: miXQuAD and MLQA-IDK.

miXQuAD Creation The miXQuAD dataset was created by combining XQuAD (which contains questions and contexts in 12 languages) with unanswerable questions from SQuAD v2.0. Since XQuAD only includes answerable questions, we enhanced it by identifying matching contexts in SQuAD v2.0 that contained unanswerable questions and aligning these with the corresponding XQuAD contexts across all languages. For the English-Questions setting, we paired English unanswerable questions with contexts in each target language. For the English-Contexts setting, we translated these unanswerable questions into the 11 non-English languages. This design ensures no data leakage, as XQuAD derives from SQuAD v2.0's dev set while models train exclusively on SQuAD v2.0's train set. The resulting miXQuAD dataset contains 12 language-specific test sets available in both English-Questions and English-Contexts configurations. In the English-Questions setting, each language maintains exactly 1,190 answerable and

882 unanswerable examples (2,072 total). In the English-Contexts setting, while answerable questions remain constant at 1,190, unanswerable question counts vary by language as shown in Table 1. The languages covered are English (en), Spanish (es), German (de), Greek (el), Russian (ru), Turkish (tr), Arabic (ar), Vietnamese (vi), Thai (th), Chinese (zh), Hindi (hi), and Romanian (ro).

	en	**ar**	**es**	**th**	**de**	**hi**	**tr**	**el**	**ro**	**vi**	**ru**	**zh**
A	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K	1.2K
U	.9K	.7K	.8K	.6K	.8K	.8K	.8K	.8K	.8K	.8K	.8K	.7K
T	2.1K	1.9K	2.0K	1.8K	2.0K	2.0K	2.0K	2.0K	2.0K	2.0K	2.0K	1.9K

Table 1: miXQuAD English-Contexts statistics. A: answerable, U: unanswerable questions. T: total. Overall, there are 24,000 questions (14,400 answerable, 9,600 unanswerable) across 12 languages.

MLQA-IDK Creation Starting from MLQA, a cross-lingual dataset containing answerable questions, we extended it to include unanswerable cases to create MLQA-IDK. Following techniques from Gautam et al. (2023), we employed two primary methods for generating unanswerable questions: entity swapping and antonym substitution (examples shown in Figure 6, Appendix A). These methods create unanswerable questions by substituting key information while preserving the overall structure and domain relevance of the original questions. We generated the unanswerable questions in English to create the English-Questions setting, and translated these questions into the other six languages to create the English-Contexts setting. Due to MLQA's incomplete overlap between questions and contexts across languages, the amount of data varies by language, as summarized in Table 2. The dataset encompasses seven languages: English (en), Spanish (es), German (de), Arabic (ar), Vietnamese (vi), Chinese (zh), and Hindi (hi). Our quality assessment of MLQA-IDK's generated unanswerable questions, performed by two of the authors on a random sample of 100 questions, achieved 95% inter-annotator agreement on both unanswerability and well-formedness. The analysis revealed a 7% noise rate, comparable to that reported in SQuAD v2.0's manual analysis (Rajpurkar et al., 2018).

	ar	**de**	**en**	**es**	**hi**	**vi**	**zh**
A	5.3K	4.5K	11.6K	5.3K	4.9K	5.5K	5.1K
U	3.7K	3.2K	12.9K	3.7K	3.5K	4.0K	3.6K
T	9.1K	7.7K	24.5K	9.0K	8.4K	9.5K	8.8K

Table 2: MLQA-IDK dataset statistics. A: answerable, U: unanswerable questions, T: total. Overall, there are 76.9K questions (46.7K A, 30.2K U) across 7 languages.

Evaluating Machine Translation All translations were performed using the Google Translate API. To ensure the accuracy of translations within our dataset, we implemented a back-translation strategy, as described in Lin et al. (2021). This involved translating the questions from foreign languages back into English. We then employed Sentence-BERT (Reimers and Gurevych, 2019) to generate embeddings for both the original and back-translated English texts, subsequently computing the cosine similarity between them. Only questions with a cosine similarity score above 0.75 were retained. This stringent validation procedure, depicted in Figure 2, guarantees the reliability of our dataset for evaluating the effectiveness of EQA systems across language pairs. All the statistics reported in this section, including those in Tables 1 and 2 concern the final versions of the corpora, after validation.

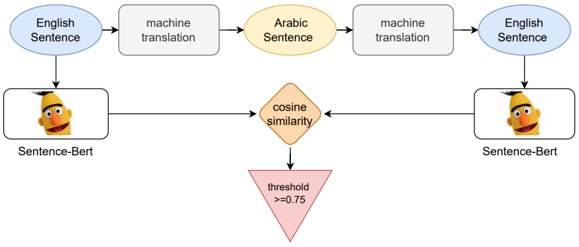


Figure 2: Example of the back-translation strategy encoding an Arabic sentence.

4 Methodology

All our experiments were conducted in a zero-shot setting, where the training data (for fine-tuning) and provided examples (for in-context learning) are exclusively in English from the SQuAD v2.0 dataset. As detailed in Appendix B, we experimented with a diverse set of models ranging from 168M to 13B parameters, using either fine-tuning or in-context learning approaches depending on model architecture and size.

4.1 Fine-Tuning Language Models

Our methodological framework centers on fine-tuning state-of-the-art multilingual language models for EQA tasks. We employ two main fine-tuning strategies to optimize model performance:

Fine-Tuning We conduct full fine-tuning on smaller multilingual models including mBERT (168M parameters), XLM-RoBERTa (279M parameters), mDeBERTa (276M parameters), and mT5-large (1.2B parameters). This approach allows us to thoroughly adapt these models to the specific requirements of cross-lingual EQA.

Parameter-Efficient Fine-Tuning For larger models like AYA-101 (13B parameters) and AYA-23 (8B parameters), we utilize two efficient fine-tuning techniques: Low-Rank Adaptation (LoRA, Hu et al., 2021) and Quantized LoRA (QLoRA, Dettmers et al., 2024). LoRA enables efficient training without fully retraining the model, while QLoRA further reduces memory usage through quantization. These methods are particularly valuable for fine-tuning large-scale LLMs while maintaining computational efficiency.

4.2 In-Context Learning

In-context learning is a pivotal methodology in our research for training models on EQA tasks, leveraging the model's ability to learn from a few examples. For this approach, we employed several large language models: GPT4-O-mini (8B parameters) using OpenAI API, Gemma-2 (9B parameters), Mistral-Nemo (12.2B parameters), AYA-101 (13B parameters), BLOOMZ (7B parameters), and AYA-23 (8B parameters). We adapted prompt formats from Slobodkin et al. (2023) to the cross-lingual case. For each prompt type, we used three different variants of few-shot prompts, each containing three examples (two answerable and one unanswerable), thus minimizing potential bias from specific example selections.

As shown in Figure 3, our approach uses two prompt families. The Question Answering family includes Regular-Prompt, Hint-Prompt, and Hint-Translate-Prompt, guiding models to provide answers or identify when no answer exists. The Hint-Prompt alerts models to potential answer absence, while Hint-Translate-Prompt adds a question translation step. The Classification family, using Answerability-Prompt, focuses solely on determining if sufficient information exists to answer the question. To identify unanswerable questions, we implemented pattern matching that includes variations of “unanswerable”, “no answer”, “unknown”, “not enough information”, and similar phrases in different contexts and formulations.

5 Results

5.1 Model Architecture and Training Approach Effects

Our analysis primarily focuses on the English-Questions setting, while a detailed discussion of the English-Contexts setting is presented in Section 5.3.1.

Encoder-only architectures Early cross-lingual EQA research relied heavily on encoder-only architectures. Table 3 reveals an intriguing characteristic of these models: while mDeBERTa achieved the best overall performance among encoders (63.64 F1), all three models exhibited remarkably strong No Answer performance (76.18-86.06 F1) but struggled significantly with answerable questions (34.13-52.26 F1). This severe imbalance suggests that encoder-only architectures excel at identifying when questions cannot be answered from the given context.

Model	Avg	Has Ans	No Ans
mBERT	56.23	34.13	**86.06**
XLM-R	58.57	45.52	76.18
mDeBERTa	**63.64**	**52.26**	78.98

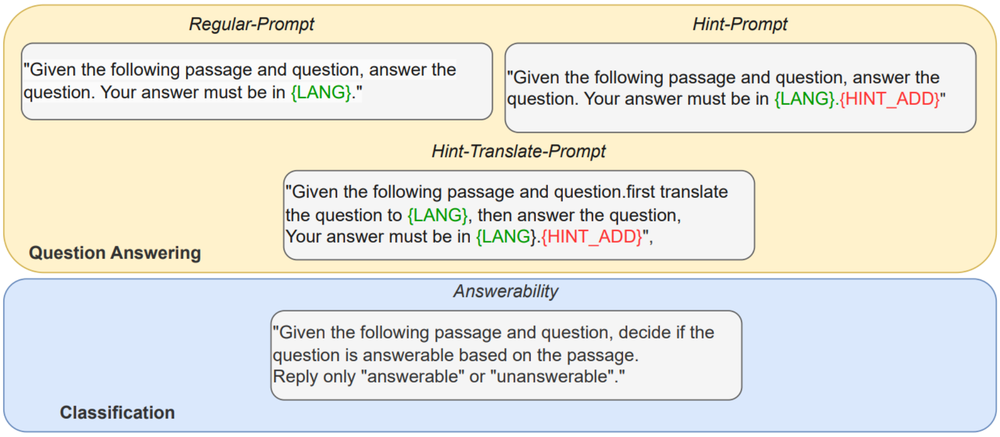
Table 3: F1 scores comparison of encoder-only models averaged across all languages in miXQuAD English-Questions setting. The bold scores represent the best performance for each category.

5.2 Model Size and Training Approach Effect

Given the limitations of encoder-only models, we focus on encoder-decoder and decoder-only architectures. Table 4 presents a comprehensive comparison revealing several key patterns in cross-lingual QA performance.

Hint prompting improves unanswerability detection. Adding hints about potential unanswerability substantially increased No Answer F1 scores across all models. AYA-101 improved from 35.16 to 54.74, while GPT4o-mini showed the most dramatic increase from 4.97 to 60.23. Crucially, this improvement maintained comparable answerable question performance (AYA-101: 67.86 vs. 66.71), suggesting that hint prompting helps models better calibrate confidence thresholds rather than simply biasing toward 'unanswerable' predictions.

Parameter count alone does not determine performance. Despite having over 10× more parameters, AYA-101 (13B) with regular prompting un-



"Given the following passage and question, answer the question. Your answer must be in {LANG}."

"Given the following passage and question, answer the question. Your answer must be in {LANG}.{HINT_ADD}"

"Given the following passage and question.first translate the question to {LANG}, then answer the question, Your answer must be in {LANG}.{HINT_ADD}",

"Given the following passage and question, decide if the question is answerable based on the passage. Reply only "answerable" or "unanswerable"."

{HINT_ADD} - "If it cannot be answered based on the passage, reply "unanswerable""

{LANG} - replace with the appropriate language

Figure 3: Example prompts used to evaluate the in-context learning methodology.

derperformed mT5-large (1.2B)—64.03 vs. 53.94 Average F1—revealing complementary strengths across architectures.[3] AYA-101 excelled at answer extraction (67.86 vs. 50.55 F1) while mT5-large dominated unanswerability detection (82.20 vs. 54.74 F1). Only through fine-tuning did AYA-101 achieve the best overall performance (81.23 F1), demonstrating that the training approach matters more than parameter count.

Verbosity correlates with poor task adaptation. Answer verbosity appears to impact only untrained large models. GPT4o-mini and Mistral-Nemo, both large models without task-specific training, produced excessively verbose answers (6-9 words on average) which likely contributed to their lower F1 scores through partial match penalties. In contrast, all other models regardless of size maintained concise responses similar to gold answer length (1-3 words), as detailed in Appendix G. This pattern suggests that excessive verbosity is a symptom of insufficient task adaptation rather than an inherent characteristic of model size.

Translation-based prompting consistently underperforms. Hint-translate prompting systematically underperformed standard hint prompting across all models and languages. This degradation was particularly severe for answerable questions—GPT4o-mini dropped from 49.60 to 30.98 F1, while even well-performing models like BLOOMZ showed similar patterns. The consistency of this effect suggests that additional translation steps introduce systematic errors rather than beneficial cross-lingual signals.

Model	Avg	Has Ans	No Ans
mT5-large	64.03	50.55	82.20
Aya-101	53.94	67.86	35.16
+Hint	61.61	66.71	54.74
+Hint-translate	61.41	66.20	54.94
+Fine tuned	**81.23**	**77.09**	**86.80**
GPT4o-mini	28.90	46.65	4.97
+Hint	54.12	49.60	60.23
+Hint-translate	43.49	30.98	60.37
Aya-23	41.01	58.98	16.77
+Hint	48.56	56.87	37.36
+Hint-translate	47.10	54.12	37.63
+Fine tuned	53.36	50.65	59.02
BLOOMZ	41.57	42.33	40.56
+Hint	48.03	37.69	61.98
+Hint-translate	47.96	39.11	59.89
Gemma-2	30.63	34.87	24.91
+Hint	38.93	40.04	37.44
+Hint-translate	38.99	43.61	32.75
Mistral-Nemo	29.96	42.59	12.91
+Hint	50.96	43.86	60.54
+Hint-translate	48.39	42.50	56.33

Table 4: Average F1 scores by category on miXQuAD English-Questions setting. The bold scores represent the best performance for each category.

5.3 Language Dependency

5.3.1 English-Questions vs. English-Contexts

In the English-Contexts setting (Table 5), hint prompting maintained its effectiveness across models. AYA-101 showed improvement from its regular prompt version. Similar enhancements were observed in other models like BLOOMZ, Gemma-2, and notably Mistral-Nemo. As shown in Figure 4, analyzing Average F1 scores across all model variants (regular, hint, hint-translate, and

---

[3] Statistical significance was assessed using t-tests between systems across all languages.

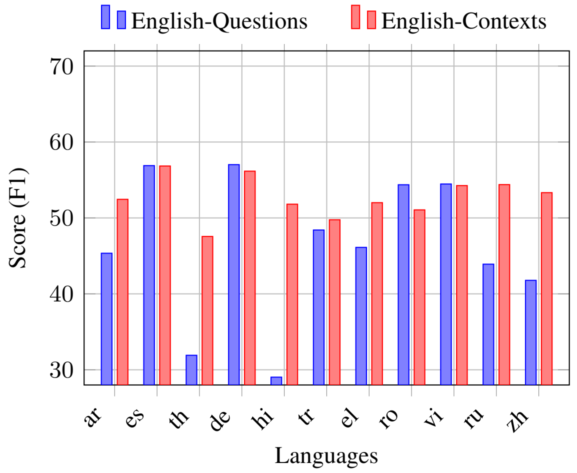


Figure 4: Average F1 scores across languages for English-Questions and English-Contexts settings. Each bar represents the mean performance of all models and their variants (regular, hint, hint-translate, and fine-tuned) for each language, comparing both experimental settings.

fine-tuned), models consistently performed better in the English-Contexts setting compared to the English-Questions setting. The gap is largest for Asian languages (Thai: 47.56 vs. 31.91, Hindi: 51.81 vs. 29.03) and the performances are comparable for European languages (Spanish: 56.84 vs 56.89, German: 56.16 vs. 57.02). The data suggests that processing questions in various languages while keeping English contexts is more manageable than handling contexts in different languages. This pattern can be attributed to two key factors: (1) questions generally have simpler linguistic structures than full contexts, making them easier to process across languages, and (2) answer extraction in English (the training data language) is more straightforward than in other languages. The complete model-specific performance can be found in Appendix B.

5.3.2 Language Similarity

The clustering methodology comprised two sequential phases, first applying K-Means clustering (k=3) to each model's F1 performance data across has-answer, no-answer, and combined metrics, then synthesizing results through a co-occurrence matrix quantifying language pair clustering frequency. Final consensus clusters were determined using hierarchical clustering with Ward's linkage (Ward Jr, 1963). The analysis reveals three distinct clusters (Figure 5): (1) five Indo-European languages (Russian, Romanian, Spanish, German, English); (2) Vietnamese, Turkish, Arabic, and Greek—languages from four different language families; and (3) Chinese, Hindi, and Thai, which all use non-Latin scripts. The second cluster's unexpected grouping of unrelated languages suggests that factors other than linguistic family relationships determine model performance patterns in cross-lingual transfer.

Model	Avg	Has Ans	No Ans
mT5-large	72.38	70.99	74.24
Aya-101	70.30	**81.07**	55.77
+Hint	73.24	79.03	65.43
+Hint-translate	72.94	79.13	64.59
+Fine tuned	**80.96**	78.44	**84.35**
GPT4o-mini	17.34	26.40	5.12
+Hint	47.42	39.48	58.15
+Hint-translate	28.52	28.99	27.87
Aya-23	51.59	72.31	23.65
+Hint	57.28	70.31	39.69
+Hint-translate	55.53	69.58	36.56
+Fine tuned	63.07	68.03	56.38
BLOOMZ	52.14	53.57	50.22
+Hint	57.54	47.66	70.88
+Hint-translate	57.79	46.99	72.35
Gemma-2	47.75	61.07	29.79
+Hint	50.23	61.02	35.67
+Hint-translate	49.49	62.20	32.34
Mistral-Nemo	30.23	44.84	10.50
+Hint	51.77	36.38	72.53
+Hint-translate	49.79	37.97	65.73

Table 5: Average F1 scores by category on miXQuAD English-Contexts setting. The bold scores represent the best performance for each category.

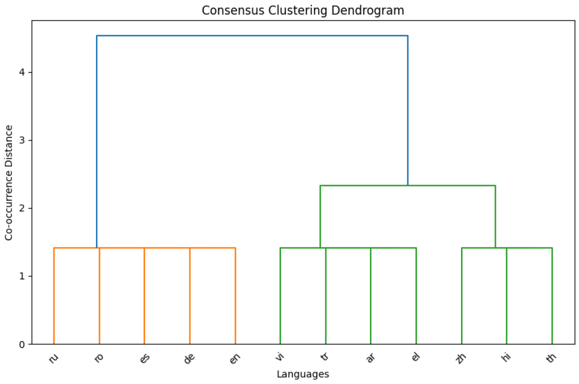


Figure 5: Hierarchical clustering dendrogram of languages based on consensus clustering across all models, using F1 scores averaged over has-answer and no-answer questions

5.4 Evaluating Model Robustness

We evaluated model robustness using MLQA-IDK beyond their original training domain. Since all models were trained on SQuAD v2.0 or received examples from it during in-context learning, this dataset provides insight into their ability to generalize to unseen data sources. Results, shown in Table 6, confirm that hint prompting improves performance across models, particularly for unanswerable questions, while fine-tuned AYA-101 maintains performance on unseen data, achieving 69.96 F1 and 76.23 F1 in English-Questions and English-Contexts respectively.

We examined model adaptability to open-domain question answering using XTREME-UP QA (Ruder et al., 2023), which includes Arabic, Finnish, Japanese, Korean, Russian, and 26 low-resource Indic languages. This dataset pairs questions with gold paragraphs for evaluation of answer extraction capabilities in multilingual settings. Results are in Table 16 (Appendix F). In this case too, hint prompting enhanced performance on unanswerable questions, improving the ability to classify cases where no answer exists in the provided context. We also observe that fine-tuning AYA-101 improved performance on unanswerable questions. The inclusion of XTREME-UP emphasizes evaluation in open-domain QA settings, particularly in low-resource language contexts.

6 Advanced Analysis

6.1 Error Pattern Analysis

We conducted a detailed analysis of models' performance as binary classifiers for question answerability across different prompting strategies. Two main approaches were evaluated: Hint-Prompt, which explicitly warns about potential unanswerable questions, and Answerability-Prompt, which focuses solely on determining answerability. As shown in Table 7, our analysis reveals a consistent trade-off across models. Hint-Prompt achieves higher recall on answerable questions but moderate recall on unanswerable ones, while Answerability-Prompt shows the opposite pattern. For example, AYA-101 with Hint-Prompt achieves 0.929/0.601 recall (answerable/unanswerable) compared to 0.706/0.849 with Answerability-Prompt. These patterns suggest that the choice between prompting strategies depends on whether correctly identifying answerable questions or detecting unanswerable ones is more critical for the specific application.

6.2 Testing Models' Reliance on Parametric Knowledge

Parametric knowledge refers to information stored in the model's parameters during pre-training that can be accessed without external context. To evaluate the extent of this knowledge in our models, we conducted two experiments. First, using regular prompts, we evaluated answerable questions without their corresponding contexts. All models showed performance drops when context was removed, with AYA-101's average F1 score falling from 73.23 to 2.63 (see in Appendix D). To further validate these findings on truly unseen data, we evaluated our models on repliQA-Trans, which was created after all models' pre-training by translating 500 answerable questions from RepliQA (Monteiro et al., 2024) into 11 languages using our validated machine translation pipeline (Section 3.2). The results show that Gemma-2 achieved the highest average performance (87.37 F1), followed by fine-tuned AYA-101 (81.87 F1) and AYA-23 (80.14 F1). The complete language-specific results are presented in Appendix E, demonstrating that models maintain performance even on post-training data.

6.3 Uncertainty Estimation

We analyzed model confidence in cross-lingual QA using the Claim Conditioned Probability method from Vashurin et al. (2025). This white-box method estimates uncertainty based on the probability of the predicted answer conditioned on the input using the model's internal logits. The evaluation included six models: AYA-23, AYA-101, BLOOMZ, and a fine-tuned variant of AYA-101, evaluated on the miXQuAD dataset. Building on our findings that hint prompting improves unanswerable question classification, we observe two key results: hint prompting reduces uncertainty when classifying unanswerable questions compared to regular prompting, aligning with the improved performance on "No answer" cases in our F1 evaluations (Section 5.2). Additionally, while fine-tuning improves overall model certainty, it reduces the confidence gap between correct and incorrect predictions—fine-tuned models show lower uncertainty overall but lose the ability to distinguish between confidence levels for right versus wrong answers compared to non-fine-tuned models. Detailed uncertainty scores and calibration analysis are in Appendix H.

7 Conclusion

Our study advances cross-lingual EQA through several key findings. Fine-tuning large language models proved most effective, with AYA-101 outperforming both smaller fine-tuned models and large models using in-context learning. Hint prompting enhanced unanswerable question detection without compromising answerable performance. Models

	English-Questions			*English-Contexts*		
Model	**Average**	**Has Ans**	**No Ans**	**Average**	**Has Ans**	**No Ans**
mT5-large	51.24	44.50	70.79	65.66	66.72	64.57
Aya-101	48.47	57.50	37.09	59.69	76.11	37.89
+Hint	52.68	56.08	48.50	65.42	73.91	54.08
+Hint-translate	52.44	55.62	48.61	65.20	73.95	53.52
+Fine tuned	**69.96**	**62.05**	**81.02**	**76.23**	**76.43**	**75.88**
GPT4o-mini	25.47	40.37	6.51	16.30	26.59	3.84
+Hint	47.20	41.87	55.41	41.52	37.79	47.50
+Hint-translate	40.99	27.91	59.42	27.55	29.67	24.59
Aya-23	36.71	56.35	11.38	49.26	73.43	17.52
+Hint	42.78	53.28	29.83	52.96	70.72	29.75
+Hint-translate	41.49	51.89	28.79	51.28	69.91	26.94
+Fine tuned	43.41	42.69	44.98	56.35	61.89	49.09
BLOOMZ	43.01	55.01	27.79	54.58	71.66	32.15
+Hint	46.42	48.18	44.83	57.79	65.28	48.05
+Hint-translate	46.77	51.89	40.40	57.94	65.22	48.49
Gemma-2	27.31	37.54	14.87	42.17	59.42	19.79
+Hint	33.11	41.13	24.00	43.09	59.54	22.07
+Hint-translate	32.65	43.98	19.08	42.87	61.20	19.38
Mistral-Nemo	29.43	41.85	13.81	30.64	45.14	12.08
+Hint	45.60	39.06	55.43	47.69	35.29	65.76
+Hint-translate	44.43	38.93	52.75	47.11	36.38	62.81

Table 6: Average F1 scores by category on MLQA-IDK comparing English-Questions and English-Contexts settings. The bold scores represent the best performance for each category.

	H-Prompt				**Answerability**			
Model	**HA**		**NA**		**HA**		**NA**	
	P	R	P	R	P	R	P	R
AYA-23	.67	.94	.83	.39	.75	.83	.73	.62
BLOOMZ	.73	.65	.59	.66	.64	.83	.61	.36
AYA-101	.76	.93	.86	.60	.86	.71	.68	.85
GPT4-O	.76	.94	.88	.59	.90	.71	.69	.89
Gemma-2	.66	.90	.72	.37	.74	.61	.58	.71
M-Nemo	.75	.76	.67	.67	.89	.35	.52	.94

Table 7: Precision (P) and Recall (R) metrics averaged across MiXQuAD test sets for question answerability. Acronyms: HA (Has Answer), NA (No Answer), H-Prompt (Hint-Prompt), M-Nemo (Mistral-Nemo), GPT4-O (GPT4o-mini).

showed better performance with English contexts versus English questions. Performance patterns aligned to some extent with traditional language families, indicating the influence of linguistic relationships. Our study also suggests that models rely on contextual understanding rather than memorized knowledge. These findings support developing more effective cross-lingual EQA systems across diverse languages.

Limitations

This work evaluated scenarios where one language is English, leaving unexplored the model's capability to handle question-answer pairs between two non-English languages. While our study focuses on datasets derived from SQuAD and MLQA, future research should explore cases of longer contexts and more complex reasoning, such as in the monolingual HotpotQA dataset. Additionally, for the use of prompting with LLMs, we only considered few-shot prompting with English examples. Future work could explore instruction-only prompting, where no examples are provided.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by grants from the Data Science Research Center at Ben-Gurion University of the Negev and the BGU/Philadelphia Academic Bridge (The Sutnick/Zipkin Endowment Fund).

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, et al. 2024. Aya 23: Open weight releases to further multilingual progress. *arXiv preprint arXiv:2405.15032*.

Akari Asai, Jungo Kasai, Jonathan H Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2020. Xor qa: Cross-lingual open-retrieval question answering. *arXiv preprint arXiv:2010.11856*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic Spanish translation of SQuAD dataset for multi-lingual question answering. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5515–5523, Marseille, France. European Language Resources Association.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

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

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Vagrant Gautam, Miaoran Zhang, and Dietrich Klakow. 2023. A lightweight method to generate unanswerable questions in English. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7349–7360, Singapore. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common sense beyond English: Evaluating and improving multilingual language models for commonsense reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1274–1287, Online. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Joao Monteiro, Pierre-Andre Noel, Etienne Marcotte, Sai Rajeswar, Valentina Zantedeschi, David Vazquez, Nicolas Chapados, Christopher Pal, and Perouz Taslakian. 2024. Repliqa: A question-answering dataset for benchmarking llms on unseen reference content. *arXiv preprint arXiv:2406.11811*.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Leonardo Ranaldi, Barry Haddow, and Alexandra Birch. 2025. Multilingual retrieval-augmented generation for knowledge-intensive task. *arXiv preprint arXiv:2504.03616*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sebastian Ruder, Jonathan H Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia, Shruti Rijhwani, Parker Riley, Jean-Michel A Sarr, Xinyi Wang, et al. 2023. Xtreme-up: A user-centric scarce-data benchmark for under-represented languages. *arXiv preprint arXiv:2305.11938*.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un) answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625.

Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

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

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Examples of Unanswerable Questions

Figure 6 shows examples of our unanswerable questions generation methods: entity swapping (questions 1-2) and antonym substitution (question 3).

B Additional Details of Models

Table 8 provides a comprehensive overview of the transformer models used in our experiments, including their architectures, parameter counts, and adaptation methods.

C Model Performance on Different Settings

In this section, we present detailed performance results across different experimental settings. Tables 9 and 10 show the full results on miXQuAD English-Questions and English-Contexts settings, respectively. While Tables 11 and 12 show results on MLQA-IDK English-Questions and English-Contexts settings, respectively.

D Analysis of Context Dependency

To investigate how models leverage contextual information versus parametric knowledge, we compare performance with and without providing the context passage. Our analysis examines the models' performance across different languages and question types when contextual information is introduced. Table 13 demonstrates the substantial performance improvements observed when contextual information is provided.

E Performance on Post-Training Data

To validate our findings on truly unseen data, we evaluate performance on repliQA-Trans, created after model pre-training. Table 14 shows Has Answer F1 scores across all languages, demonstrating the models' ability to generalize to new content.

F Performance on Open-Domain Data

We evaluated our models as answer extraction components for open-domain QA systems using XTREME-UP QA's reading comprehension data (Ruder et al., 2023), where questions are paired with gold paragraphs. Table 16 shows detailed performance across languages, while the full dataset statistics can be found in Table 15, demonstrating the models' effectiveness at extracting answers from retrieved documents.

G Answer Length Analysis

To analyze the verbosity of model responses, we compared the average number of words in model predictions with gold answers across different settings. Table 17 presents this analysis for both MiXQuAD and MLQA-IDK datasets in English-Questions and English-Contexts settings. Notably, larger models like GPT4o-mini and Mistral-Nemo tend to produce longer answers, with averages up to 3 times the length of gold answers in some settings.

H Uncertainty Estimation Analysis

This section provides detailed results from our uncertainty estimation analysis using the Claim Conditioned Probability method. The uncertainty score is calculated as:

$$\text{Uncertainty} = \log P(\text{claim}|\text{input}) \qquad (1)$$

where $P(\text{claim}|\text{input})$ is the probability of the model's predicted answer (claim) given the input question and context. This method leverages the model's internal logits to estimate confidence in the prediction.

Table 18 presents uncertainty scores for correctly classified unanswerable questions across different models and prompting strategies.

Context:

As of 2005, recombinant growth hormones available in the United States (and their manufacturers) included Nutropin (Genentech), Humatrope (Lilly), Genotropin (Pfizer), Norditropin (Novo), and Saizen (Merck Serono). In 2006, the U.S. Food and Drug Administration (FDA) approved a version of rHGH called Omnitrope (Sandoz). A sustained-release form of growth hormone, Nutropin Depot (Genentech and Alkermes) was approved by the FDA in 1999, allowing for fewer injections (every 2 or 4 weeks instead of daily); however, the product was discontinued by Genentech/Alkermes in 2004 for financial reasons.

Unanswerable Questions:

1. *[Entity Swap]* What company manufactured the first Genotropin approved rHGH?
2. *[Entity Swap]* What company manufactured the first FDA approved Omnitrope?
3. *[Antonym]* What company manufactured the last FDA approved rHGH?

Figure 6: Example of unanswerable questions from MLQA-IDK. While the questions appear answerable and are related to the context, they require information beyond what is provided in the passage.

Model	**Arch**	**Params**	**FT**	**ICL**
mBERT (Devlin et al., 2019)	E	168M	✓	x
XLM-RoBERTa (Conneau et al., 2020)	E	279M	✓	x
mDeBERTa (He et al., 2020)	E	276M	✓	x
mT5-large (Xue et al., 2021)	E-D	1.2B	✓	x
GPT4o-mini (Hurst et al., 2024)	D	8B	x	✓
Gemma-2 (Team et al., 2024)	D	9B	x	✓
Mistral-Nemo (Sreenivas et al., 2024)	D	12.2B	x	✓
AYA-101 (Üstün et al., 2024)	E-D	13B	✓	✓
BLOOMZ (Muennighoff et al., 2023)	D	7B	x	✓
Aya-23 (Aryabumi et al., 2024)	D	8B	✓	✓

Table 8: Overview of Transformer Models and Methodologies. Arch: Architecture (E: Encoder, D: Decoder); Params: Number of parameters; FT: Fine-Tuning; ICL: In-Context Learning. FT and ICL represent our proposed methods for adapting models to the G-XLT task. ✓ indicates the method was applied to the model.

The results demonstrate that hint prompting consistently reduces uncertainty (increases confidence) when correctly classifying unanswerable questions across most models. Notably, hint prompting also increases the number of correctly classified unanswerable instances, supporting our main findings about improved performance on "No answer" cases. Table 19 compares uncertainty scores between correct and incorrect predictions to assess model calibration. Well-calibrated models should exhibit higher confidence (lower uncertainty) for correct predictions compared to incorrect ones.

I Hyperparameters

All experiments were conducted using NVIDIA RTX 6000 Ada GPUs for LLMs (AYA-101, AYA-23, BLOOMZ) and RTX 4090 GPUs for encoder-only models (mBERT, XLM-RoBERTa, mDeBERTa) and mT5-large.

I.1 Encoder-Only Models

For fine-tuning the encoder-only models mBERT[4], XLM-RoBERTa[5], and mDeBERTa[6], we utilized a

---

[4] `https://huggingface.co/bert-base-multilingual-cased`

[5] `https://huggingface.co/xlm-roberta-base`

[6] `https://huggingface.co/microsoft/mdeberta-v3-base`

Category	Model\Language	ar	es	th	de	hi	tr	el	ro	vi	en	ru	zh	Avg
Average	**mT5-large**	52.64	72.72	59.29	70.78	57.70	63.35	56.06	72.26	63.96	84.19	56.97	58.40	64.03
	AYA-101	51.85	63.35	26.63	66.77	25.12	55.45	58.83	63.40	58.55	74.23	52.89	50.23	53.94
	+Hint	58.52	72.09	34.98	73.10	33.29	63.11	65.76	72.58	66.77	80.22	59.70	59.18	61.61
	+Hint-translate	58.04	71.11	35.17	72.79	34.18	62.52	65.75	71.89	66.34	79.90	59.31	59.86	61.41
	+Fine tuned	**78.42**	**84.10**	**81.39**	**83.54**	**81.96**	**77.35**	**80.55**	**82.91**	**79.03**	**88.43**	**78.85**	**78.17**	**81.23**
	GPT4o-mini	31.82	34.56	9.80	37.61	6.75	30.80	34.30	31.70	38.56	40.22	28.97	21.76	28.90
	+Hint	55.02	62.58	34.49	64.53	33.58	57.64	59.87	61.31	63.37	66.89	48.97	41.24	54.12
	+Hint-translate	45.27	47.27	32.69	50.80	28.30	43.93	47.93	45.20	52.00	60.84	35.21	32.44	43.49
	Aya-23	40.71	51.58	13.03	53.13	14.66	43.68	40.20	50.23	48.40	65.76	38.49	32.23	41.01
	+Hint	48.07	61.67	21.69	60.68	20.41	52.46	46.99	58.25	55.73	70.53	46.40	39.88	48.56
	+Hint-translate	45.41	59.69	24.60	58.12	21.28	50.49	44.23	56.97	54.13	69.23	43.85	37.15	47.10
	+Fine tuned	49.04	54.20	42.02	65.69	36.09	53.13	52.07	54.82	58.51	79.00	50.70	45.06	53.36
	BLOOMZ	44.89	55.71	29.97	43.39	19.00	34.93	31.88	39.52	51.13	73.74	29.55	45.15	41.57
	+Hint	50.54	60.39	38.83	50.35	24.37	42.99	40.80	46.03	55.63	77.17	38.37	50.88	48.03
	+Hint-translate	50.81	61.25	38.61	50.17	22.77	42.93	40.86	45.80	55.94	77.10	38.39	50.87	47.96
	Gemma-2	22.07	39.05	22.74	40.17	20.55	28.54	23.13	37.68	32.92	62.22	20.00	18.45	30.63
	+Hint	32.21	49.80	29.55	49.80	27.57	38.04	31.06	46.67	43.70	62.08	29.85	26.85	38.93
	+Hint-translate	31.02	50.39	28.00	50.01	26.65	39.89	30.44	47.41	44.98	60.50	31.42	27.16	38.99
	Mistral-Nemo	22.32	33.57	11.70	40.15	12.90	29.50	26.82	38.21	37.32	59.15	27.78	20.06	29.96
	+Hint	43.57	57.80	29.10	59.67	32.12	53.39	48.44	59.45	58.54	72.87	53.48	43.09	50.96
	+Hint-translate	40.13	51.83	25.86	56.24	30.47	52.55	42.46	59.51	58.46	70.97	53.00	39.18	48.39
Has answer	**mT5-large**	30.15	66.28	43.90	62.05	37.95	51.65	36.85	65.47	52.88	82.89	35.32	41.26	50.55
	AYA-101	**74.76**	**81.59**	24.65	**81.35**	20.62	71.11	**78.40**	**81.66**	**77.68**	**85.35**	**74.24**	62.90	67.86
	+Hint	74.47	80.15	24.60	79.01	20.57	70.45	76.86	80.52	76.71	83.22	72.68	61.22	66.71
	+Hint-translate	73.99	79.25	24.42	78.48	20.55	69.84	76.45	79.88	77.08	83.26	72.07	59.15	66.20
	+Fine tuned	73.09	81.15	**78.40**	80.03	**79.32**	**71.86**	76.67	80.47	75.08	84.76	72.61	**71.68**	**77.09**
	GPT4o-mini	51.53	58.70	17.06	56.24	11.75	51.60	58.80	54.40	56.25	64.14	49.81	29.49	46.65
	+Hint	52.98	63.03	17.57	59.03	13.43	55.11	59.75	58.98	58.69	70.54	51.81	34.29	49.60
	+Hint-translate	34.66	40.84	12.46	33.29	7.88	32.87	30.97	33.23	38.19	56.90	34.24	16.18	30.98
	Aya-23	62.54	76.48	16.89	71.64	18.43	61.91	60.64	73.98	72.17	84.75	58.42	49.87	58.98
	+Hint	60.64	71.52	16.70	67.75	18.64	58.88	59.97	70.47	68.78	83.22	57.14	48.76	56.87
	+Hint-translate	56.62	70.20	16.14	66.52	17.75	56.67	53.45	68.65	66.34	83.43	51.48	42.14	54.12
	+Fine tuned	52.68	67.74	4.07	61.78	6.76	55.63	57.05	65.38	65.92	77.13	56.76	36.94	50.65
	BLOOMZ	61.01	61.56	3.89	46.93	20.17	20.59	15.82	40.74	73.54	83.55	21.23	58.90	42.33
	+Hint	53.31	54.06	2.53	41.42	19.53	14.87	11.73	35.44	69.66	77.47	17.87	54.34	37.69
	+Hint-translate	62.44	56.37	2.61	41.08	19.98	14.91	11.78	34.72	73.68	76.42	17.57	57.78	39.11
	Gemma-2	16.40	45.61	17.79	51.35	16.64	39.19	26.05	43.90	40.85	74.89	21.45	24.28	34.87
	+Hint	23.50	49.51	19.07	55.45	17.36	47.60	30.21	49.53	48.97	77.70	28.38	33.18	40.04
	+Hint-translate	25.76	58.43	19.33	58.43	17.10	52.41	31.67	55.75	55.68	78.83	31.90	38.07	43.61
	Mistral-Nemo	36.00	52.40	18.71	55.22	14.78	43.43	45.15	53.83	47.05	78.56	41.45	24.53	42.59
	+Hint	41.29	56.06	19.13	56.31	14.97	45.60	52.93	54.44	40.30	75.17	44.71	25.38	43.86
	+Hint-translate	44.25	53.35	18.02	50.53	13.61	46.61	47.98	51.62	39.80	73.88	42.38	27.97	42.50
No answer	**mT5-large**	83.00	81.41	80.05	82.54	84.36	79.14	81.98	81.41	78.92	85.95	86.17	81.52	82.20
	AYA-101	20.94	38.74	29.29	47.09	31.18	34.32	32.43	38.78	32.73	59.23	24.08	33.15	35.16
	+Hint	37.00	61.23	48.98	65.12	50.46	53.22	50.80	61.87	53.37	76.16	42.18	56.43	54.74
	+Hint-translate	36.51	60.13	49.66	65.12	52.57	52.65	51.33	61.12	51.86	75.36	42.11	60.81	54.94
	+Fine tuned	**85.61**	**88.06**	**85.42**	**88.29**	**85.53**	**84.77**	**85.79**	**86.21**	**84.36**	**93.39**	**87.27**	**86.93**	**86.80**
	GPT4o-mini	5.22	1.99	0.00	12.48	0.00	2.73	1.25	1.08	14.69	7.94	0.86	11.34	4.97
	+Hint	57.77	61.97	57.32	71.94	60.78	61.06	60.04	64.46	69.68	61.97	45.13	50.63	60.23
	+Hint-translate	59.59	55.96	59.98	74.44	55.84	58.85	70.81	61.34	70.64	66.16	36.51	54.37	60.37
	Aya-23	11.27	17.99	7.83	28.16	9.57	19.09	12.63	18.18	16.33	40.14	11.61	8.43	16.77
	+Hint	31.11	48.38	28.43	51.14	22.79	43.81	29.48	41.77	38.14	53.41	31.90	27.90	37.36
	+Hint-translate	30.28	45.51	36.02	46.79	26.04	42.14	31.79	41.20	37.65	50.08	33.57	30.43	37.63
	+Fine tuned	46.04	38.67	93.13	73.17	76.08	54.05	48.61	39.80	52.01	81.30	43.88	61.49	59.02
	BLOOMZ	23.13	47.81	65.16	38.63	17.43	54.28	53.56	37.87	20.90	60.51	40.78	26.61	40.56
	+Hint	46.79	68.94	87.80	62.40	30.88	80.92	80.01	60.32	36.70	76.76	66.03	46.23	61.98
	+Hint-translate	35.11	67.84	87.19	62.44	26.54	80.73	80.09	60.74	32.02	78.01	66.48	41.54	59.89
	Gemma-2	29.71	30.20	29.41	25.10	25.82	14.18	19.20	29.29	22.23	45.13	18.03	10.59	24.91
	+Hint	43.96	50.19	43.69	42.18	41.35	25.14	32.20	42.82	36.59	41.01	31.83	18.30	37.44
	+Hint-translate	38.10	39.54	39.69	38.67	39.54	22.98	28.77	36.17	30.54	35.76	30.77	12.44	32.75
	Mistral-Nemo	3.86	8.17	2.23	19.81	10.36	10.70	2.08	17.13	24.19	32.96	9.34	14.03	12.91
	+Hint	46.64	60.13	42.56	64.22	55.26	63.91	42.37	66.22	83.15	69.77	65.31	66.97	60.54
	+Hint-translate	34.59	49.78	36.44	63.95	53.22	60.55	35.00	70.15	83.64	67.05	67.31	54.31	56.33

Table 9: F1 scores (averaged over 3 model seeds) across three categories (Average F1, Has answer F1, and No answer F1) on miXQuAD, English-Questions across 12 language pairs. The best performance for each category and language appears in bold.

learning rate of 5e-5, a batch size of 12, and a maximum sequence length of 384. A document stride of 128 was applied to handle overlapping text segments. Training was conducted over 2 epochs with a weight decay of 0.01 using the AdamW optimizer. The inputs were lowercased before processing to ensure consistency.

I.2 mT5-large Fine-tuning

For fine-tuning the mT5-large model[7] on SQuAD 2.0, we used a learning rate of 5e-5 and a training batch size of 2. The model was trained for 5 epochs with a maximum sequence length of 512. AdamW was employed as the optimizer to facilitate effective gradient updates during fine-tuning.

---

[7] `https://huggingface.co/google/mt5-large`

Category	Model\Language	ar	es	th	de	hi	tr	el	ro	vi	en	ru	zh	Avg
Average	**mT5-large**	66.29	76.19	71.35	75.19	69.72	69.83	72.41	74.71	65.92	84.19	73.77	68.96	72.38
	AYA-101	61.68	68.03	69.26	74.73	69.68	71.86	70.71	72.43	69.81	74.23	71.65	69.52	70.30
	+Hint	69.10	75.06	71.75	75.72	72.31	73.42	72.92	73.77	71.28	80.22	72.93	70.38	73.24
	+Hint-translate	68.73	74.74	71.81	75.45	71.79	72.96	72.55	73.41	71.17	79.90	72.50	70.26	72.94
	+Fine tuned	**78.46**	**83.12**	**79.69**	**84.07**	**78.94**	**79.06**	**79.74**	**81.59**	**77.50**	**88.43**	**81.21**	**79.66**	**80.96**
	GPT4o-mini	9.66	13.74	22.22	17.57	13.65	18.49	12.01	14.87	17.80	40.22	12.57	15.33	17.34
	+Hint	39.97	45.25	52.56	48.41	42.69	42.07	46.08	43.95	49.71	66.89	46.28	45.23	47.42
	+Hint-translate	24.17	30.45	22.63	29.24	24.06	26.30	23.31	25.37	24.42	60.84	25.78	25.61	28.52
	Aya-23	51.45	56.17	28.27	57.88	46.65	52.51	52.90	53.00	51.18	65.76	53.64	49.71	51.59
	+Hint	56.41	62.27	34.52	64.28	52.31	57.60	58.51	59.30	57.05	70.53	59.65	54.89	57.28
	+Hint-translate	54.38	59.67	35.03	62.62	50.10	55.82	56.52	57.88	54.90	69.23	57.82	52.33	55.53
	+Fine tuned	62.53	65.57	43.52	68.50	58.22	61.33	64.78	57.88	62.19	78.90	65.65	67.75	63.07
	BLOOMZ	56.19	63.78	33.02	51.88	57.45	41.25	40.04	38.32	59.42	73.74	48.46	62.16	52.14
	+Hint	61.29	69.66	41.60	56.83	63.55	43.77	44.92	45.34	64.04	77.17	54.61	67.72	57.54
	+Hint-translate	61.99	69.92	41.77	56.57	63.94	43.49	44.96	45.89	64.84	77.10	55.08	67.89	57.79
	Gemma-2	50.66	48.81	50.20	47.84	45.09	38.74	49.66	43.66	46.13	62.22	48.93	41.09	47.75
	+Hint	52.44	52.20	53.15	50.84	47.60	40.13	53.39	48.44	47.85	62.08	51.58	43.05	50.23
	+Hint-translate	51.87	51.83	51.31	50.50	47.33	39.00	52.69	47.97	46.26	60.50	51.91	42.73	49.49
	Mistral-Nemo	24.02	27.21	28.66	29.79	24.92	25.53	27.21	22.58	36.70	59.15	31.20	25.73	30.23
	+Hint	50.21	51.20	50.24	51.69	46.06	47.12	49.74	47.05	50.92	72.87	53.67	50.47	51.77
	+Hint-translate	49.99	48.71	46.24	49.68	41.92	44.67	47.25	44.81	50.36	70.97	53.28	49.54	49.79
Has answer	**mT5-large**	60.21	75.76	69.68	73.02	73.99	70.91	71.20	74.36	62.84	82.89	71.46	65.61	70.99
	AYA-101	**79.91**	**82.63**	**79.36**	**81.64**	**79.28**	**79.97**	**82.78**	**82.07**	**80.52**	**85.35**	**81.34**	**77.99**	**81.07**
	+Hint	77.46	80.09	77.98	79.34	77.58	77.84	81.55	80.37	78.50	83.22	79.02	75.37	79.03
	+Hint-translate	77.56	79.96	78.23	79.61	77.51	77.94	81.65	80.30	78.59	83.26	79.29	75.66	79.13
	+Fine tuned	73.84	80.29	77.52	82.35	77.64	77.18	76.76	80.25	75.60	84.76	77.45	77.69	78.44
	GPT4o-mini	14.55	21.05	34.65	24.16	21.15	28.46	17.55	23.44	26.45	64.14	18.74	22.49	26.40
	+Hint	28.58	33.15	46.85	36.98	40.96	37.08	34.86	33.62	41.68	70.54	32.31	37.15	39.48
	+Hint-translate	22.84	28.47	24.07	29.60	27.26	27.90	24.32	25.68	27.40	56.90	26.10	27.36	28.99
	Aya-23	73.80	78.42	41.06	76.65	66.71	72.15	78.54	72.87	72.84	84.75	77.79	72.10	72.31
	+Hint	71.86	76.43	37.85	74.30	63.26	69.16	76.66	71.03	71.96	83.22	76.83	71.11	70.31
	+Hint-translate	70.73	74.65	40.32	73.31	61.90	68.58	75.75	70.07	70.60	83.43	75.86	69.80	69.58
	+Fine tuned	69.32	74.17	44.39	71.23	66.03	65.38	72.01	70.52	69.30	77.13	70.28	66.58	68.03
	BLOOMZ	72.87	80.60	3.63	57.72	75.88	8.07	15.26	34.84	74.49	83.55	60.96	74.93	53.57
	+Hint	67.32	75.71	1.31	48.24	71.32	4.08	9.14	26.28	68.20	77.47	54.63	68.19	47.66
	+Hint-translate	67.15	75.12	1.46	45.62	71.02	3.43	8.78	24.49	68.01	76.42	54.67	67.72	46.99
	Gemma-2	61.25	55.68	61.28	52.73	60.24	57.65	67.34	54.85	61.47	74.89	66.01	59.42	61.07
	+Hint	59.91	57.32	61.62	53.20	58.76	56.85	66.43	54.65	61.04	77.70	64.67	60.09	61.02
	+Hint-translate	62.47	58.61	62.30	55.20	59.72	57.14	67.43	55.25	61.99	78.83	66.06	61.39	62.20
	Mistral-Nemo	40.42	41.27	48.77	42.70	40.92	38.51	45.89	34.07	44.51	78.56	43.76	38.69	44.84
	+Hint	33.94	37.27	49.67	33.36	36.80	26.82	35.82	21.22	21.12	75.17	28.29	37.09	36.38
	+Hint-translate	34.44	38.26	51.18	34.08	39.09	29.46	38.52	23.40	21.84	73.88	31.36	40.10	37.97
No answer	**mT5-large**	74.49	76.76	73.59	78.12	63.95	68.37	74.04	75.18	70.07	85.95	76.88	73.47	74.24
	AYA-101	37.08	48.34	55.64	65.42	56.73	60.93	54.43	59.42	55.37	59.23	58.58	58.09	55.77
	+Hint	57.83	68.26	63.35	70.83	65.20	67.47	61.27	64.86	61.53	76.16	64.71	63.65	65.43
	+Hint-translate	56.81	67.69	63.16	69.85	64.06	66.26	60.28	64.10	61.15	75.36	63.35	62.97	64.59
	+Fine tuned	**84.70**	**86.93**	**82.62**	**86.40**	**80.69**	**81.60**	**83.75**	**83.41**	**80.05**	**93.39**	**86.29**	**82.32**	**84.35**
	GPT4o-mini	3.07	3.86	5.45	8.68	3.52	5.05	4.54	3.29	6.13	7.94	4.26	5.67	5.12
	+Hint	55.33	61.57	60.27	63.84	45.02	48.81	61.23	57.88	60.55	61.97	65.14	56.13	58.15
	+Hint-translate	25.97	33.11	20.70	28.75	19.73	24.15	21.94	24.95	20.41	66.16	25.35	23.25	27.87
	Aya-23	21.28	26.16	11.00	32.54	19.58	26.01	18.30	26.20	21.96	40.14	21.06	19.51	23.65
	+Hint	35.57	43.16	30.01	50.76	37.53	41.99	34.02	43.47	36.93	53.41	36.48	33.00	39.69
	+Hint-translate	32.32	39.46	27.90	48.19	34.17	38.59	30.58	41.43	33.72	50.08	33.49	28.77	36.56
	+Fine tuned	53.37	53.97	42.33	64.82	47.70	55.86	55.03	40.82	52.61	81.30	59.42	69.32	56.38
	BLOOMZ	33.68	41.09	72.68	44.00	32.58	86.02	73.47	43.01	39.08	60.51	31.60	44.94	50.22
	+Hint	53.14	61.49	95.96	68.41	53.07	97.32	93.20	71.06	58.43	76.76	54.58	67.09	70.88
	+Hint-translate	55.03	62.89	96.15	71.36	54.39	97.55	93.77	74.76	60.55	78.01	55.64	68.11	72.35
	Gemma-2	36.36	39.54	35.27	41.24	24.65	13.23	25.82	28.58	25.44	45.13	25.89	16.37	29.79
	+Hint	42.37	45.28	41.73	47.66	32.54	17.58	35.79	40.07	30.05	41.01	33.91	20.07	35.67
	+Hint-translate	37.57	42.67	36.48	44.15	30.62	14.52	32.81	38.14	25.02	35.76	32.81	17.54	32.34
	Mistral-Nemo	1.89	8.24	1.52	12.36	3.33	8.02	2.01	7.07	26.16	32.96	14.25	8.24	10.50
	+Hint	72.15	70.00	51.03	76.42	58.55	74.49	68.52	81.90	91.12	69.77	87.91	68.52	72.53
	+Hint-translate	70.98	62.82	39.57	70.72	45.73	65.20	59.04	73.70	88.86	67.05	82.85	62.29	65.73

Table 10: F1 scores (averaged over 3 model seeds) across three categories (Overall F1, Has answer F1, and No answer F1) on miXQuAD English-Contexts setting across 12 language pairs. The bold scores represent the best performance for each category and language.

I.3 In-Context Learning Settings

In-context learning experiments were conducted using AYA-101[8], AYA-23[9], BLOOMZ[10], GPT4o-mini (via OpenAI API), Gemma-2[11], and Mistral-Nemo[12]. The models were configured with a maximum of 25 new tokens and an input window size of 4024. To provide the models with sufficient contextual information, we employed three few-shot examples. The hint prompt template used was, "If it cannot be answered based on the passage, reply

---

[8] https://huggingface.co/CohereForAI/AYA-101

[9] https://huggingface.co/CohereForAI/aya-23

[10] https://huggingface.co/bigscience/bloomz

[11] https://huggingface.co/google/gemma-2b

[12] https://huggingface.co/mistralai/Nemo-2.0

Category	Model\Language	ar	de	en	es	hi	vi	zh	Avg
Average	**mT5-large**	46.10	52.55	65.94	55.54	46.41	49.77	42.38	51.24
	AYA-101	44.62	59.27	59.91	56.91	22.72	52.01	43.84	48.47
	+Hint	47.86	60.24	67.89	62.85	27.22	56.05	46.62	52.68
	+Hint-translate	47.31	60.19	67.71	62.63	27.48	55.24	46.51	52.44
	+Fine tuned	**67.57**	**66.87**	**81.64**	**72.48**	**71.70**	**70.80**	**58.63**	**69.96**
	GPT4o-mini	27.84	32.98	30.11	32.21	6.28	33.68	15.19	25.47
	+Hint	47.80	55.64	55.62	55.30	28.49	54.93	32.65	47.20
	+Hint-translate	42.20	47.60	51.19	45.96	25.42	48.15	26.38	40.99
	Aya-23	35.95	44.96	51.08	45.72	12.39	44.89	21.98	36.71
	+Hint	41.78	51.74	55.72	53.18	17.03	51.32	28.71	42.78
	+Hint-translate	40.04	50.09	54.60	52.15	17.50	49.34	26.74	41.49
	+Fine tuned	39.13	51.09	59.26	45.49	33.84	47.23	27.82	43.41
	BLOOMZ	45.86	39.56	55.11	48.42	18.50	51.71	41.89	43.01
	+Hint	48.01	43.84	59.85	51.00	22.79	54.17	45.30	46.42
	+Hint-translate	50.19	43.67	59.99	51.72	20.97	55.07	45.75	46.77
	Gemma-2	18.60	34.44	46.39	33.14	14.13	30.75	13.69	27.31
	+Hint	27.17	40.39	45.56	42.06	19.55	38.24	18.79	33.11
	+Hint-translate	26.01	39.96	44.46	42.59	18.56	38.97	18.01	32.65
	Mistral-Nemo	23.05	38.45	48.37	32.57	11.58	34.44	17.52	29.43
	+Hint	39.78	51.96	61.09	50.52	28.08	51.62	36.14	45.60
	+Hint-translate	38.28	51.03	60.27	49.10	27.51	52.56	32.26	44.43
Has answer	**mT5-large**	26.53	49.23	78.65	54.61	30.17	51.83	20.50	44.50
	AYA-101	**60.40**	**64.84**	78.17	**72.16**	17.83	**66.54**	**42.58**	**57.50**
	+Hint	59.89	62.56	76.33	69.43	17.59	64.88	41.87	56.08
	+Hint-translate	59.61	62.07	76.41	69.32	17.36	64.66	39.88	55.62
	+Fine tuned	58.04	57.85	**80.42**	67.33	**66.71**	63.23	40.78	62.05
	GPT4o-mini	44.19	48.23	57.77	53.60	10.72	48.85	19.21	40.37
	+Hint	44.92	46.11	65.18	53.74	11.89	49.03	22.22	41.87
	+Hint-translate	30.21	27.07	51.39	37.35	7.35	33.67	8.32	27.91
	Aya-23	56.65	63.10	82.64	71.18	16.62	69.99	34.25	56.35
	+Hint	54.89	58.27	80.62	64.42	16.53	66.07	32.14	53.28
	+Hint-translate	52.67	57.87	80.88	64.24	15.88	63.51	28.20	51.89
	+Fine tuned	41.60	47.13	66.69	60.19	7.02	54.21	21.96	42.69
	BLOOMZ	62.87	39.11	77.70	55.58	20.94	76.97	51.90	55.01
	+Hint	54.13	30.66	71.13	46.23	19.64	71.84	43.60	48.18
	+Hint-translate	65.06	31.34	71.14	48.76	20.55	76.69	49.67	51.89
	Gemma-2	18.61	50.41	72.48	44.49	13.45	44.61	18.70	37.54
	+Hint	24.94	52.19	74.79	47.47	13.85	51.41	23.28	41.13
	+Hint-translate	26.39	54.46	76.16	54.83	13.87	56.42	25.73	43.98
	Mistral-Nemo	36.44	51.44	74.28	50.57	13.00	45.85	21.40	41.85
	+Hint	38.93	46.23	69.99	50.23	12.63	36.54	18.86	39.06
	+Hint-translate	42.84	42.52	69.01	49.24	12.37	35.36	21.16	38.93
No answer	**mT5-large**	73.98	67.14	69.20	67.87	76.73	67.25	73.39	70.79
	AYA-101	22.05	51.46	43.57	35.33	29.61	31.97	45.62	37.09
	+Hint	30.65	56.98	60.34	53.55	40.75	43.86	53.37	48.50
	+Hint-translate	29.73	57.56	59.92	53.17	41.70	42.25	55.94	48.61
	+Fine tuned	**81.20**	**79.51**	**82.73**	**79.76**	**78.73**	**81.23**	**83.97**	**81.02**
	GPT4o-mini	4.45	11.58	5.34	1.93	0.03	12.77	9.49	6.51
	+Hint	51.92	69.00	47.07	57.51	51.85	63.08	47.46	55.41
	+Hint-translate	59.33	76.43	51.01	58.14	50.85	68.11	52.04	59.42
	Aya-23	6.35	19.51	22.82	9.69	6.46	10.27	4.53	11.38
	+Hint	23.04	42.56	33.43	37.27	17.74	30.97	23.83	29.83
	+Hint-translate	21.97	39.17	31.07	35.05	19.78	29.80	24.67	28.79
	+Fine tuned	35.60	56.65	52.60	24.69	71.55	37.61	36.15	44.98
	BLOOMZ	21.54	40.20	34.88	38.30	15.08	16.87	27.67	27.79
	+Hint	39.27	62.32	49.76	57.76	27.23	29.80	47.70	44.83
	+Hint-translate	28.94	60.97	50.00	55.90	21.56	25.26	40.18	40.40
	Gemma-2	18.60	12.04	23.04	17.08	15.09	11.63	6.58	14.87
	+Hint	30.37	23.82	19.39	34.40	27.56	20.07	12.42	24.00
	+Hint-translate	25.47	19.62	16.07	25.27	25.15	14.91	7.04	19.08
	Mistral-Nemo	3.90	20.22	25.17	7.11	9.58	18.71	12.01	13.81
	+Hint	41.00	59.99	53.13	50.94	49.81	72.43	60.70	55.43
	+Hint-translate	31.76	62.99	52.44	48.91	48.81	76.29	48.05	52.75

Table 11: F1 scores (averaged over 3 model seeds) across three categories (Overall F1, Has answer F1, and No answer F1) on MLQA-IDK English-Questions setting across 7 language pairs. The bold scores represent the best performance for each category and language.

“unanswerable”," as illustrated in Figure 3.

I.4 AYA-101 Fine-tuning

To fine-tune AYA-101 on the SQuAD 2.0 dataset, we employed the QLoRA method to optimize memory efficiency and maintain high performance. The input window size was set to 2048 tokens to accommodate longer context passages effectively. The training process used a learning rate of 3e-5 and a batch size of 2 for both training and evaluation. The LoRA-specific parameters included a rank of 64, a scaling factor (`alpha`) of 32, and a dropout rate of 0.1 to prevent overfitting. The LoRA bias was configured as “none ”, and all linear layers in the model were targeted for parameter-efficient updates.

Category	Model\Language	ar	de	en	es	hi	vi	zh	Avg
Average	**mT5-large**	61.12	68.09	73.66	69.06	63.41	61.67	62.64	65.66
	AYA-101	56.29	67.44	59.91	59.56	57.93	57.32	59.35	59.69
	+Hint	62.30	68.76	67.89	65.44	63.74	63.90	65.91	65.42
	+Hint-translate	62.08	68.45	67.71	65.32	63.52	63.73	65.60	65.20
	+Fine tuned	**73.03**	**78.05**	**81.64**	**77.64**	**73.86**	**73.53**	**75.85**	**76.23**
	GPT4o-mini	10.06	17.56	30.11	13.58	12.31	16.84	13.65	16.30
	+Hint	36.40	42.71	55.62	39.31	35.77	42.09	38.72	41.52
	+Hint-translate	21.26	26.48	51.19	24.58	21.52	22.78	25.02	27.55
	Aya-23	48.81	54.51	51.08	53.06	43.96	48.33	45.07	49.26
	+Hint	51.99	58.36	55.72	57.07	47.04	52.23	48.32	52.96
	+Hint-translate	50.38	57.26	54.60	55.56	44.63	50.33	46.19	51.28
	+Fine tuned	54.82	59.72	59.26	58.24	51.19	52.94	58.26	56.35
	BLOOMZ	55.67	46.67	55.11	57.45	53.63	55.93	57.57	54.58
	+Hint	58.77	49.66	59.85	60.42	56.68	59.00	60.16	57.79
	+Hint-translate	58.86	49.58	59.99	60.46	57.12	59.21	60.37	57.94
	Gemma-2	44.63	42.94	46.39	43.29	39.60	42.08	36.24	42.17
	+Hint	45.92	44.25	45.56	44.98	41.18	42.46	37.29	43.09
	+Hint-translate	45.75	44.25	44.46	45.25	41.25	41.51	37.63	42.87
	Mistral-Nemo	25.70	31.13	48.37	27.46	24.24	33.80	23.80	30.64
	+Hint	43.94	48.08	61.09	46.46	42.34	47.67	44.24	47.69
	+Hint-translate	44.39	47.16	60.27	45.62	40.95	47.19	44.16	47.11
Has answer	**mT5-large**	55.81	69.47	78.65	70.37	68.78	61.60	62.33	66.72
	AYA-101	73.70	80.73	78.17	76.38	75.88	74.82	73.08	76.11
	+Hint	71.54	79.10	76.33	74.01	73.73	72.38	70.29	73.91
	+Hint-translate	71.42	79.33	76.41	74.05	73.65	72.46	70.33	73.95
	+Fine tuned	**70.17**	**78.90**	**80.42**	**76.41**	**77.11**	**75.73**	**76.25**	**76.43**
	GPT4o-mini	15.19	26.43	57.77	21.01	19.20	26.13	20.43	26.59
	+Hint	27.41	37.03	65.18	31.23	35.46	36.28	31.91	37.79
	+Hint-translate	21.12	28.84	51.39	25.78	26.79	26.54	27.24	29.67
	Aya-23	72.49	77.45	82.64	76.95	65.78	71.23	67.46	73.43
	+Hint	69.83	74.51	80.62	73.79	61.48	69.40	65.44	70.72
	+Hint-translate	69.11	73.81	80.88	73.38	60.18	68.21	63.82	69.91
	+Fine tuned	60.61	64.78	66.69	65.29	59.70	58.89	57.30	61.89
	BLOOMZ	74.94	52.51	77.70	76.22	73.86	73.53	72.83	71.66
	+Hint	69.08	43.93	71.13	70.70	68.97	67.66	65.50	65.28
	+Hint-translate	69.11	42.99	71.14	70.73	69.30	67.62	65.62	65.22
	Gemma-2	59.10	56.26	72.48	56.01	57.52	60.03	54.53	59.42
	+Hint	58.47	56.04	74.79	56.95	56.18	59.57	54.77	59.54
	+Hint-translate	60.88	58.09	76.16	58.62	57.63	60.54	56.46	61.20
	Mistral-Nemo	40.39	43.98	74.28	41.89	39.29	42.51	33.67	45.14
	+Hint	28.87	29.95	69.99	34.42	34.06	19.56	30.19	35.29
	+Hint-translate	29.03	30.30	69.01	35.12	37.08	20.58	33.51	36.38
No answer	**mT5-large**	68.73	66.14	69.20	67.19	55.85	61.78	63.08	64.57
	AYA-101	31.39	48.80	43.57	35.76	32.69	33.18	39.85	37.89
	+Hint	49.08	54.25	60.34	53.31	49.70	52.20	59.69	54.08
	+Hint-translate	48.72	53.19	59.92	52.97	49.28	51.70	58.86	53.52
	+Fine tuned	**77.12**	**76.85**	**82.73**	**79.38**	**69.28**	**70.50**	**75.28**	**75.88**
	GPT4o-mini	2.71	5.12	5.34	3.08	2.61	4.03	4.02	3.84
	+Hint	49.25	50.67	47.07	50.75	36.21	50.11	48.41	47.50
	+Hint-translate	21.47	23.18	51.01	22.89	14.10	17.59	21.87	24.59
	Aya-23	14.94	22.33	22.82	19.26	13.27	16.75	13.24	17.52
	+Hint	26.48	35.71	33.43	33.40	26.73	28.55	23.98	29.75
	+Hint-translate	23.59	34.02	31.07	30.35	22.76	25.67	21.14	26.94
	+Fine tuned	46.54	52.63	52.60	48.25	39.21	44.74	59.63	49.09
	BLOOMZ	28.11	38.47	34.88	30.88	25.17	31.66	35.89	32.15
	+Hint	44.04	57.69	49.76	45.87	39.39	47.05	52.57	48.05
	+Hint-translate	44.19	58.84	50.00	45.93	39.99	47.61	52.90	48.49
	Gemma-2	23.95	24.25	23.04	25.28	14.41	17.32	10.25	19.79
	+Hint	27.97	27.71	19.39	28.05	20.08	18.86	12.45	22.07
	+Hint-translate	24.11	24.84	16.07	26.32	18.20	15.26	10.87	19.38
	Mistral-Nemo	4.70	13.08	25.17	7.03	3.06	21.78	9.77	12.08
	+Hint	65.50	73.52	53.13	63.51	53.98	86.44	64.21	65.76
	+Hint-translate	66.35	70.81	52.44	60.48	46.39	83.89	59.29	62.81

Table 12: F1 scores (averaged over 3 model seeds) across three categories (Overall F1, Has answer F1, and No answer F1) on MLQA-IDK English-Contexts setting across 7 language pairs. The bold scores represent the best performance for each category and language.

The model was fine-tuned over 2 epochs using the AdamW8bit optimizer, which supports low-memory operations while ensuring efficient gradient updates. Additionally, a weight decay of 0.01 was applied to regularize the model and prevent overfitting. This configuration enabled effective fine-tuning of AYA-101 on the question-answering task while optimizing for both memory and computational efficiency.

To further optimize memory and computational efficiency, we employed 4-bit quantization.

I.5 Fine-tuning AYA-23

For fine-tuning AYA-23 on SQuAD 2.0, we used a learning rate of 3e-5 and set the batch size to 2 for both training and evaluation. The LoRA-

Model	ar	es	th	de	hi	tr	el	ro	vi	en	ru	zh	Avg
Aya-23	7.08	10.43	0.78	13.63	5.76	9.44	10.31	10.32	11.66	24.07	12.20	6.73	10.20
+context	68.19	74.80	38.72	73.87	61.53	67.91	76.18	67.23	66.73	83.77	73.97	64.12	68.09
BLOOMZ	0.38	0.23	0.00	0.00	0.17	0.00	0.00	0.00	0.41	0.66	0.10	0.23	0.18
+context	70.82	77.73	2.71	56.62	73.13	7.63	14.51	31.70	72.74	82.04	59.74	73.92	51.94
AYA-101	2.53	2.57	4.28	1.24	2.14	4.17	0.88	4.26	2.13	1.24	4.41	1.73	2.63
+context	70.00	73.74	70.50	74.71	70.16	72.79	76.65	74.20	73.90	79.01	73.65	69.46	73.23
Gemma-2	8.42	9.78	8.31	10.70	6.43	9.13	11.12	11.24	10.97	16.00	11.00	8.07	10.10
+context	58.40	51.93	58.94	49.76	59.93	55.92	64.21	52.23	58.79	72.10	63.29	60.17	58.81
Mistral-Nemo	10.65	8.32	9.69	10.31	8.31	7.35	10.86	6.70	10.50	20.87	8.83	8.05	10.04
+context	41.43	41.70	50.95	43.34	43.41	40.26	48.76	36.02	46.62	79.19	45.51	40.59	46.48

Table 13: Has Answer Performance Across Languages: No Context vs. With Context

Model\Language	ar	es	th	de	hi	tr	el	ro	vi	en	ru	zh	Avg
AYA-101	71.29	66.32	59.13	69.23	71.20	69.61	72.33	71.39	64.37	62.86	70.79	68.80	68.11
+hint	71.83	67.43	58.84	69.25	71.53	70.88	73.21	72.44	65.68	62.14	71.95	69.11	68.69
+hint translate	71.40	66.61	58.58	68.50	69.87	70.52	72.57	71.78	64.99	61.57	71.15	67.83	67.95
+fine tuned	85.44	84.08	60.93	80.36	**85.34**	78.28	85.29	84.12	83.10	88.64	83.54	83.27	81.87
GPT4o-mini	36.37	27.06	41.24	34.91	45.50	40.25	28.99	34.50	36.97	58.15	40.91	33.27	38.18
+hint	70.73	56.94	63.76	64.16	78.90	65.65	66.74	61.00	62.95	85.18	70.92	60.88	67.32
+hint translate	15.73	19.85	17.75	22.44	19.89	19.32	16.87	21.86	19.22	57.78	18.39	18.31	22.28
Aya-23	79.60	87.96	45.25	88.00	67.94	81.45	89.81	**86.67**	78.37	95.07	85.91	75.68	80.14
+hint	79.33	84.99	44.87	87.51	64.36	80.36	88.45	85.06	77.48	95.38	84.22	73.59	78.80
+hint translate	74.75	81.04	46.41	87.19	64.19	78.68	85.19	82.96	75.21	95.37	82.14	72.10	77.10
BLOOMZ	82.27	86.27	16.76	66.57	84.23	34.52	35.27	55.91	81.01	89.88	73.54	**83.67**	65.83
+hint	79.50	84.53	11.03	62.67	82.58	24.64	26.39	51.20	77.89	87.91	69.26	81.49	61.59
+hint translate	76.68	82.99	9.17	58.66	79.98	21.77	23.03	47.42	76.28	86.16	64.95	78.47	58.80
Gemma-2	**91.06**	**88.42**	**86.67**	**89.19**	83.67	**84.49**	**92.34**	82.13	**85.06**	96.02	85.97	83.41	**87.37**
+hint	90.48	88.13	86.12	88.90	81.70	83.18	90.76	82.70	83.84	**96.33**	85.98	80.55	86.56
+hint translate	90.42	87.53	85.30	89.49	83.32	83.73	90.68	83.35	83.62	96.41	**87.10**	82.13	86.92
Mistral-Nemo	62.10	60.34	60.03	66.30	56.51	66.10	68.34	62.70	66.21	94.42	64.27	54.08	65.12
+hint	60.38	58.86	60.12	53.15	56.01	57.84	62.81	58.15	46.60	94.68	57.41	55.49	60.13
+hint translate	55.40	53.06	52.50	47.55	50.69	51.40	57.08	53.91	43.50	94.03	54.16	52.34	55.47

Table 14: Has Answer F1 scores across 12 language pairs on repliQA-Trans dataset (English-Contexts setting). The bold scores represent the best performance for each category and language.

Category	ar	fi	ja	ko	ru	avg_indi
Answered (A)	301	312	240	282	235	538
Unanswered (U)	281	294	231	266	213	96
Total	582	606	471	548	448	634

Table 15: XTREME-UP QA dataset statistics on English context settings. A: answerable questions, U: unanswerable questions, Total: total number of questions. The **avg_indi** column represents the average over 26 low-resource languages included in XTREME-UP (Ruder et al., 2023).

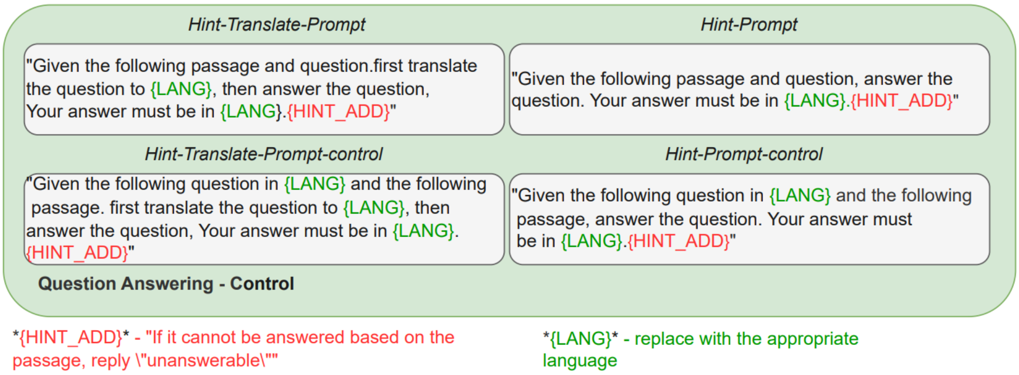
specific parameters included a rank of 64, an alpha value of 32, and a dropout rate of 0.1, with the bias set to "none ". All linear layers were targeted for parameter-efficient fine-tuning. The model was trained for 3 epochs using the AdamW optimizer with a weight decay of 0.01, and the maximum sequence length was configured to 2048 tokens.

J Prompt Control Analysis

We conducted control experiments to evaluate model sensitivity to prompt variations, particularly focusing on language specification placement and minor prompt changes. As shown in Figure 7, we tested four prompt variants: Hint-Prompt, Hint-Translate-Prompt, and their respective control versions that explicitly mention the target language in the question description. Results in Table 20 show minimal performance differences between these variants, indicating that the model is robust to such prompt modifications.

Category	Model\Language	ar	fi	ja	ko	ru	avg_indi	avg
Average	**mT5-large**	59.66	59.23	67.11	65.36	70.78	50.77	53.30
	AYA-101	73.20	72.12	**73.19**	73.09	73.63	**65.99**	**67.30**
	+hint	**73.47**	71.30	72.89	72.91	74.33	63.90	65.58
	+hint translate	73.39	72.09	72.80	72.72	**74.72**	63.89	65.60
	+fine tuned	72.57	68.03	72.81	**75.22**	70.60	52.90	56.41
	GPT4o-mini	10.87	11.95	13.60	15.47	9.44	20.54	19.01
	+hint	60.25	56.37	62.40	63.64	54.19	44.80	47.50
	+hint translate	20.78	28.82	33.20	45.17	32.77	19.00	21.44
	Aya-23	50.44	48.92	45.35	54.34	53.29	39.49	41.53
	+hint	61.80	60.98	60.45	66.14	64.39	40.08	44.28
	+hint translate	57.45	60.26	55.49	63.44	60.43	40.67	44.14
	+fine tuned	61.86	54.94	68.42	62.10	64.26	40.40	44.46
	BLOOMZ	68.44	-	65.91	53.41	60.81	52.61	53.86
	+hint	67.63	-	64.55	52.27	62.11	47.69	49.80
	+hint translate	68.03	-	64.47	52.75	62.21	47.66	49.81
	Gemma-2	23.25	10.69	22.44	22.99	25.37	22.06	21.86
	+hint	31.70	40.34	46.34	42.86	40.53	26.97	29.45
	+hint translate	35.52	42.97	46.92	41.57	46.30	30.06	32.39
Has answer	**mT5-large**	54.71	41.96	59.83	59.99	58.98	47.30	48.74
	AYA-101	65.67	**56.22**	56.83	63.41	66.60	**62.04**	**61.99**
	+hint	63.53	53.33	55.12	60.12	65.81	58.74	58.90
	+hint translate	63.95	54.44	54.94	60.11	67.13	58.85	59.09
	+fine tuned	54.60	46.12	55.59	59.47	52.45	45.46	46.98
	GPT4o-mini	18.03	19.67	20.43	24.03	16.29	23.09	22.46
	+hint	30.44	22.94	32.45	37.50	19.89	34.84	33.69
	+hint translate	11.60	16.55	15.14	20.39	13.95	13.23	13.65
	Aya-23	**70.40**	53.14	57.04	**74.62**	**71.22**	41.66	46.04
	+hint	67.77	42.27	56.41	68.36	67.28	34.09	38.97
	+hint translate	65.44	44.61	54.73	70.56	66.40	35.81	40.35
	+fine tuned	67.34	34.15	**60.93**	62.98	65.62	39.27	42.77
	BLOOMZ	57.58	-	50.19	19.87	49.54	47.74	45.53
	+hint	45.27	-	39.17	11.38	41.94	38.85	36.80
	+hint translate	45.37	-	38.88	12.32	42.15	38.69	36.70
	Gemma-2	44.06	19.69	40.01	43.02	45.66	27.15	29.25
	+hint	39.69	13.93	36.21	40.25	36.55	21.06	23.33
	+hint translate	42.54	19.25	37.07	40.34	41.03	24.28	26.46
No answer	**mT5-large**	64.95	77.56	74.68	71.06	83.81	69.43	70.35
	AYA-101	81.26	89.01	90.19	83.34	81.38	85.50	85.41
	+hint	84.11	90.37	91.35	86.47	83.73	89.06	88.72
	+hint translate	83.52	90.82	91.35	86.10	83.10	88.54	88.25
	+fine tuned	91.82	91.27	90.70	91.92	90.62	88.52	89.03
	GPT4o-mini	3.21	3.75	6.50	6.40	1.88	6.84	6.38
	+hint	**92.18**	**91.84**	**93.51**	91.36	**92.02**	90.62	**90.91**
	+hint translate	30.61	41.84	51.95	71.43	53.53	43.01	44.28
	Aya-23	29.07	44.45	33.19	32.84	33.49	31.88	32.39
	+hint	55.40	80.84	64.65	63.79	61.19	71.71	70.50
	+hint translate	48.88	76.88	56.28	55.89	53.84	67.31	65.65
	+fine tuned	56.00	76.99	76.20	61.16	62.76	47.41	50.97
	BLOOMZ	80.08	-	82.26	88.98	73.24	77.16	78.50
	+hint	91.58	-	90.91	**95.62**	84.36	90.31	87.71
	+hint translate	90.29	-	91.06	**95.62**	84.36	**90.76**	86.11
	Gemma-2	0.95	1.14	4.19	1.76	2.98	1.15	1.34
	+hint	23.14	68.37	56.86	45.62	44.92	59.53	57.35
	+hint translate	28.00	68.15	57.15	42.86	52.12	61.52	59.32

Table 16: F1 scores across three categories (Overall F1, Has answer F1, and No answer F1) on XTREME-UP dataset with English-Contexts setting. The table shows performance across different languages with various models. The bold scores represent the best performance for each category and language. The **avg_indi** column represents the average over 26 low-resource languages included in XTREME-UP.


Hint-Translate-Prompt
Hint-Prompt
Hint-Translate-Prompt-control
Hint-Prompt-control
Question Answering - Control

"Given the following passage and question.first translate the question to {LANG}, then answer the question, Your answer must be in {LANG}.{HINT_ADD}"

"Given the following passage and question, answer the question. Your answer must be in {LANG}.{HINT_ADD}"

"Given the following question in {LANG} and the following passage. first translate the question to {LANG}, then answer the question, Your answer must be in {LANG}.{HINT_ADD}"

"Given the following question in {LANG} and the following passage, answer the question. Your answer must be in {LANG}.{HINT_ADD}"

{HINT_ADD} - "If it cannot be answered based on the passage, reply \"unanswerable\""

{LANG} - replace with the appropriate language

Figure 7: Prompt variations used in control experiments to test model robustness to language specification and prompt formatting.

Model	**MiXQuAD English-Questions**	**MiXQuAD English-Contexts**	**MLQA-IDK English-Questions**	**MLQA-IDK English-Contexts**
mBERT	1.36	1.58	1.51	1.45
XLM-R	1.80	1.69	1.74	1.53
mDeBERTa	1.99	2.47	1.89	2.21
mT5-large	1.91	2.43	1.79	2.39
AYA-23	3.58	3.52	3.52	3.60
+Hint	3.25	3.36	3.19	3.45
+Hint-translate	3.34	3.44	3.27	3.51
+Fine tuned	1.86	2.21	1.69	2.01
BLOOMZ	1.94	1.84	2.01	2.16
+Hint	1.56	1.48	1.70	1.91
+Hint-translate	1.69	1.51	1.95	2.00
AYA-101	2.61	2.34	2.54	2.05
+Hint	2.50	2.13	2.28	1.46
+Hint-translate	2.56	2.14	2.36	1.44
+Fine tuned	2.30	2.42	2.01	2.32
GPT4o-mini	6.08	9.38	6.92	9.41
+Hint	4.78	7.48	4.95	7.26
+Hint-translate	7.25	11.36	6.67	9.38
Gemma-2	2.43	2.34	2.33	2.28
+Hint	2.39	2.52	2.34	2.52
+Hint-translate	2.72	2.73	2.69	2.75
Mistral-Nemo	4.77	6.88	4.69	6.49
+Hint	3.61	3.80	3.38	3.39
+Hint-translate	3.97	4.59	3.74	3.85
Average (gold)	**2.86**	**2.92**	**3.31**	**3.18**

Table 17: Average number of words in model predictions compared to gold answers across development sets in MiXQuAD and MLQA-IDK datasets for both English-Questions and English-Contexts settings.

Model	Method	Count	Uncertainty
AYA-23	Regular-Prompt	4034	-0.8205
AYA-23	Hint-Prompt	8939	-0.8802
AYA-101	Regular-Prompt	11752	-0.8738
AYA-101	Hint-Prompt	12347	-0.8917
BLOOMZ	Regular-Prompt	8309	-0.6782
BLOOMZ	Hint-Prompt	13351	-0.7840

Table 18: Uncertainty scores for correctly classified unanswerable questions across models and prompting methods. Lower uncertainty values (more negative) indicate higher confidence. Count represents the number of questions where the model predicted "unanswerable" and the true label was also "unanswerable" (true positives for the unanswerable category).

Model	Method	Correct	Incorrect	Diff.
AYA-23	Regular-Prompt	-0.8492	-0.7198	-0.1295
AYA-23	Hint-Prompt	-0.8571	-0.7256	-0.1315
AYA-101	Regular-Prompt	-0.9001	-0.8288	-0.0713
AYA-101	Hint-Prompt	-0.9087	-0.8273	-0.0814
AYA-101	Fine-Tuned	-0.8652	-0.8751	0.0099
BLOOMZ	Regular-Prompt	-0.7713	-0.6675	-0.1038
BLOOMZ	Hint-Prompt	-0.8168	-0.7108	-0.1060

Table 19: Mean uncertainty scores for correct versus incorrect predictions. Negative differences indicate higher confidence for correct predictions. The fine-tuned AYA-101 model shows reduced calibration with near-zero difference.

Category	Model\Language	ar	es	th	de	hi	tr	el	ro	vi	en	ru	zh	Avg
Average	hint	**55.49**	**71.42**	35.69	**71.73**	**31.65**	**62.50**	**62.01**	**68.66**	**54.86**	**79.02**	59.50	58.09	**59.22**
	hint-control	55.12	70.54	**35.84**	70.91	30.29	62.13	61.42	67.92	53.22	78.79	**59.61**	57.55	58.61
	hint-translate	54.62	71.06	35.39	70.60	31.31	61.36	60.43	66.74	53.10	78.63	58.94	**58.12**	58.36
	hint-translate-control	54.15	68.96	35.35	70.13	30.89	61.09	58.46	65.52	51.85	78.33	58.41	57.41	57.55
Has answer	hint	**71.31**	**76.79**	**23.73**	**76.65**	**19.23**	**68.74**	**71.66**	**75.60**	**63.59**	80.53	70.90	59.13	**63.16**
	hint-control	71.10	76.43	23.65	75.57	17.77	68.27	71.14	74.81	60.56	80.20	**70.94**	**59.18**	62.47
	hint-translate	70.40	76.16	23.72	74.69	17.62	67.16	70.18	73.76	61.36	**80.68**	70.36	56.48	61.88
	hint-translate-control	69.66	75.52	23.56	73.96	17.23	66.86	67.33	72.83	60.36	80.50	70.10	57.10	61.25
No answer	hint	**34.13**	**64.18**	**51.82**	**65.08**	**48.42**	**54.09**	**48.98**	**59.30**	43.09	**76.99**	44.11	56.69	**53.91**
	hint-control	33.57	62.59	52.27	64.63	47.17	53.86	48.30	58.62	**43.32**	76.88	**44.34**	55.33	53.41
	hint-translate	33.34	**64.18**	51.14	**65.08**	**49.78**	53.52	47.28	57.26	41.96	75.86	43.54	**60.32**	53.61
	hint-translate-control	33.22	60.10	51.25	64.97	49.32	53.29	46.49	55.67	40.37	75.40	42.64	57.83	52.55

Table 20: F1 scores across three categories (Average F1, Has Answer F1, and No Answer F1) for AYA-101 on miXQuAD in the English-Questions setting across 12 language pairs. The bold scores represent the best performance for each category and language.

Evaluating Compositional Generalisation in VLMs and Diffusion Models

Beth Pearson
University of Bristol
`beth.pearson@bristol.ac.uk`

Bilal Boulbarss
University of Amsterdam
`bilal.boulbarss@student.uva.nl`

Michael Wray
University of Bristol
`michael.wray@bristol.ac.uk`

Martha Lewis
University of Amsterdam
`m.a.f.lewis@uva.nl`

Abstract

A fundamental aspect of the semantics of natural language is that novel meanings can be formed from the composition of previously known parts. Vision-language models (VLMs) have made significant progress in recent years, however, there is evidence that they are unable to perform this kind of composition. For example, given an image of a red cube and a blue cylinder, a VLM such as CLIP is likely to incorrectly label the image as a red cylinder or a blue cube, indicating it represents the image as a 'bag-of-words' and fails to capture compositional semantics. Diffusion models have gained significant attention for their impressive generative abilities, and zero-shot classifiers based on diffusion models have been shown to perform competitively with CLIP in certain compositional tasks. In this work we explore whether the generative Diffusion Classifier has improved compositional generalisation abilities compared to discriminative models. We assess three models—Diffusion Classifier, CLIP, and ViLT—on their ability to bind objects with attributes and relations in both zero-shot learning (ZSL) and generalised zero-shot learning (GZSL) settings. Our results show that the Diffusion Classifier and ViLT perform well at concept binding tasks, but that all models struggle significantly with the relational GZSL task, underscoring the broader challenges VLMs face with relational reasoning. Analysis of CLIP embeddings suggests that the difficulty may stem from overly similar representations of relational concepts such as left and right. Code and dataset are available at: github.com/otmive/diffusion_classifier_clip

1 Introduction

Compositionality is a fundamental part of how humans learn (Chomsky, 1957; Janssen and Partee, 1997). It allows us to take familiar concepts and combine them in new ways to interpret novel situations, learn from limited examples, and build increasingly complex ideas. Within formal semantics as in e.g. (Montague, 1973), compositionality is assumed in the formalism. However, vision-language models (VLMs) fall short in tasks requiring compositional understanding (Diwan et al., 2022; Yuksekgonul et al., 2022; Lewis et al., 2024). Even with advances in attention mechanisms (Vani et al., 2024) and positional encoding (Su et al., 2024), VLMs are unable to match the compositional reasoning skills of humans (Sinha et al., 2024; Hua et al., 2024b). VLMs such as CLIP (Radford et al., 2021) have been shown to treat captions as a bag-of-words (Thrush et al., 2022) and are not able to bind concepts to objects in the same way humans can. For example, given an image of a *red cube* and a *blue cylinder*, a VLM may misinterpret the image as containing a *blue cube* or a *red cylinder* (see Figure 1). Additionally, a VLM should be able to generalise learned concepts to new unseen combinations of attributes and objects: if a model learns the colour *cyan* through images of *cyan cone* and the shape *cube* through images of *green cubes*, it should also be able to recognise images of *cyan cubes* or *green cones*. In formal semantics, given *correct* representations of the words *green*, *cyan*, *cube*, and *cone*, this property would naturally occur.

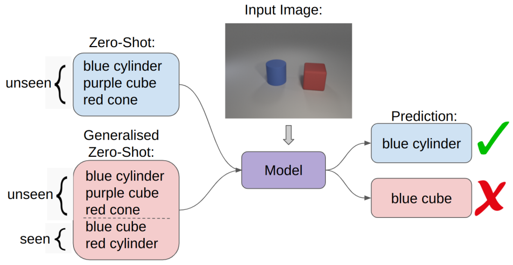


Figure 1: We evaluate the compositional generalisation of Vision-Language Models (VLMs) by assessing their ability to bind colours to objects and relations to objects in both zero-shot and generalised zero-shot settings across single-object, two-object, and relational scenarios

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 122–133
November 8-9, 2025 ©2025 Association for Computational Linguistics

Diffusion Models have gained significant interest in recent years for their state-of-the-art performance on image generation (Ramesh et al., 2022; Dhariwal and Nichol, 2021) and editing tasks (Brooks et al., 2023). Their performance as zero-shot classifiers in vision tasks is a recent topic of exploration (Clark and Jaini, 2023; Krojer et al., 2023). On compositional benchmarks such as Winoground (Thrush et al., 2022) or the Concept Binding Benchmark from Lewis et al. (2024), their performance has been shown to be comparable to that of CLIP (Li et al., 2023; Clark and Jaini, 2023). However, Winoground has been argued to require commonsense and world knowledge rather than purely testing for compositional abilities (Diwan et al., 2022), and performance on the Concept Binding Benchmark can be at chance.

In this paper, we contribute to the understanding of the compositional abilities of diffusion model-based classifiers by comparing with transformer-based classifiers on compositional tasks. Specifically, we explore how these two types of models are able to compose attributes and relations—tasks VLMs particularly struggle with. We aim to assess whether Diffusion Classifier can offer new insights or improvements in handling these challenging aspects of compositional semantics.

We consider two settings for our experiments—zero-shot learning (ZSL) and generalised zero-shot learning (GZSL). In ZSL, the aim is to recognise only unseen classes whereas GZSL aims to train models that are able to discriminate between both seen and unseen classes during test time (Pourpanah et al., 2022; Xian et al., 2017). The GZSL setting is particularly important for real world scenarios as there may only be labelled data for a small number of classes and capturing every possible class in the training set is often impossible. Therefore, it is important for models to be able to generalise to unseen classes in the presence of labels that have previously been seen.

To probe these abilities, we extend the Concept Binding Benchmark from Lewis et al. (2024), which evaluates model performance on attribute-object binding and relational composition. We evaluate the performance of Diffusion Classifier—a classifier built from Stable Diffusion (Rombach et al., 2022)—comparing it with CLIP and ViLT (Kim et al., 2021). Despite the dataset being lightweight, it still proves challenging for the models, particularly in the important GZSL setting.

The main contributions of this work are threefold: **(1)** We compare Diffusion Classifier, CLIP, and ViLT on compositional tasks. Diffusion Classifier generalises best in single-object settings, however, ViLT has by far the best two-object performance. All models struggle to reliably compose relations with objects. **(2)** To provide a more robust evaluation of compositional generalisation, we present our extension of the Concept Binding Benchmark from Lewis et al. (2024). This extended benchmark consists of three datasets to test VLMs in both zero-shot learning (ZSL) and generalised zero-shot learning (GZSL) scenarios. **(3)** We analyse the effects of fine-tuning on compositional semantic understanding, showing that models fail to form correct representations for spatial relations.

2 Related Work

Benchmarking Compositionality in VLMs There is a growing interest in the ability of VLMs to reason compositionally, with several benchmarks being proposed in recent years (Yuksekgonul et al., 2022; Ma et al., 2023; Hsieh et al., 2024; Dumpala et al., 2024; Ray et al., 2024; Zhao et al., 2022; Huang et al., 2024; Thrush et al., 2022; Hua et al., 2024a). Compositional generalisation is an important ability for VLMs to have because it encourages the interpretability and data efficiency of models (Bommasani et al., 2021). However, it has been argued (Lewis et al., 2024; Hsieh et al., 2024) that various compositionality benchmarks are ‘hackable’, showing that in some cases it is possible to solve the benchmark simply by comparing prompts (Wu et al., 2023) and ignoring the image. SugarCrepe (Hsieh et al., 2024) is designed to deal with this problem, but is still prone to the issue that the correct caption is statistically more likely in the training corpus. Unlike benchmarks that use complex real-world images, we use simple, synthetic images to ensure no spurious correlations and to directly test compositional understanding. We argue that VLMs should be able to handle these simpler reasoning tasks before advancing to more complex, real-world images.

Improving Compositionality in VLMs Methods have been proposed to improve the compositional abilities of VLMs (Cascante-Bonilla et al., 2023; Doveh et al., 2023). Several works use hard negative sampling to fine-tune CLIP on batches of similar images e.g. “a black cat sitting on a desk” and “a black desk sitting on a cat” which force the model to learn more detailed representations of the data (Yuksekgonul et al., 2022; Shou and Lin, 2024; Sahin et al., 2024). Other methods include different representations for objects within images such as trees or graphs (Singh et al., 2023; Yellinek et al., 2025) and adaptations to the contrastive loss function of CLIP to include more compositional supervision (Pandey et al., 2023; Zhang et al., 2024). Despite advancements, VLMs still struggle with compositional reasoning (Hsieh et al., 2024; Dumpala et al., 2024). Our benchmark aims to investigate why VLMs struggle with compositional tasks by testing in GZSL settings using in-distribution and out-of-distribution images to identify potential biases.

Diffusion Model Classifiers Recently, methods have been proposed to leverage diffusion models as zero-shot classifiers (Chen et al., 2023; Li et al., 2023; Krojer et al., 2023; Clark and Jaini, 2023). Li et al. (2023) propose Diffusion Classifier, a model built from Stable Diffusion, which achieves a higher accuracy than CLIP on tasks requiring compositional reasoning such as concept binding. Krojer et al. (2023) use a similar method for using Stable Diffusion (Rombach et al., 2022) as a classifier but include a normalising value based on the noise prediction error calculated with no text guidance. He et al. (2023) use the attention scores between the image and text representations of Stable Diffusion to adapt it for image-text matching tasks. Clark and Jaini (2023) also propose a zero-shot classifier created from Google's Imagen, which shows some ability to bind attributes such as shape, size and colour where CLIP fails to do so. For our experiments we use the Diffusion Classifier from Li et al. (2023) as Stable Diffusion is open source with easily accessible fine-tuning methods.

3 Experiments

We base the design of our benchmark on the experiments from Lewis et al. (2024) where three datasets were created for exploring composition of attributes and relations with objects. While this setup reveals that models often struggle even with simple object compositions, our aim is to extend this evaluation to include both Zero-Shot Learning (ZSL) and Generalised Zero-Shot Learning (GZSL) settings. To enable this, we adapt and expand the original benchmark to support systematic and rigorous testing in both settings.

The images are generated using the generation script for the CLEVR dataset (Johnson et al., 2017)—using a Blender script (Community, 2018) to render 3D shapes. The original code included only three shapes *cubes*, *cylinders*, and *spheres* which we extend with an additional shape, *cones*, to increase the diversity across the dataset splits. For the single and two-object datasets, we consider the following colours: *blue*, *brown*, *cyan*, *gray*, *green*, *purple*, *red*, and *yellow*. We define the label sets for the single and two-object datasets as follows:

Let C be the set of colours and S the set of shapes. For object classification, each object is identified by its colour–shape pair, and the label set is defined as:

$$\mathcal{Y} = \{(c, s) \mid c \in C,\ s \in S\}.$$

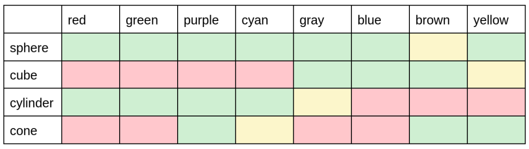
Each element of $\mathcal{Y}$ represents a unique object (e.g., *red square*, *blue circle*). In the two-object dataset, labels consist of two such tuples, e.g., $((c_1, s_1), (c_2, s_2))$. For the relational dataset, we define a set of spatial relations $R = \{\textit{left}, \textit{right}\}$. We exclude the relations *front* and *behind* which were included in Lewis et al. (Lewis et al., 2024) as we found these to be too ambiguous—distinguishing which shape is further forward is often difficult even for humans. The relational label set is then defined as:

$$\mathcal{Y}_{\text{rel}} = \{(s_i, r, s_j) \mid s_i, s_j \in S,\ s_i \neq s_j,\ r \in R\},$$

where each triple describes a relation between two distinct shapes—for example, $(\textit{circle}, \textit{left}, \textit{square})$. All datasets are partitioned into five subsets: training ($\mathcal{Y}^{\text{train}}$), in-distribution validation/test ($\mathcal{Y}^{\text{IDval}}, \mathcal{Y}^{\text{IDtest}}$), and out-of-distribution validation/test ($\mathcal{Y}^{\text{OODval}}, \mathcal{Y}^{\text{OODtest}}$). In-distribution subsets use the same label space as the training set, i.e. $\mathcal{Y}^{\text{train}} = \mathcal{Y}^{\text{IDval}} = \mathcal{Y}^{\text{IDtest}}$, while OOD splits are defined such that:

$$\mathcal{Y}^{\text{train}} \cap \mathcal{Y}^{\text{OODval}} = \emptyset, \quad \mathcal{Y}^{\text{train}} \cap \mathcal{Y}^{\text{OODtest}} = \emptyset,$$
$$\mathcal{Y}^{\text{OODval}} \cap \mathcal{Y}^{\text{OODtest}} = \emptyset.$$

This setup enables evaluation both within the training distribution and on novel combinations, to assess generalisation. We give the structure of our single and two-object datasets within Figure 2. The label *red cube* is in the test set, meaning that it is not seen during training, but *red* (e.g. in *red sphere*) and *cube* (e.g. in *gray cube*) have both been seen during training in other combinations. The structure of the relational dataset is given in Figure 3.



	red	green	purple	cyan	gray	blue	brown	yellow
sphere	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩
cube	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟨
cylinder	🟩	🟩	🟩	🟩	🟨	🟥	🟥	🟥
cone	🟥	🟥	🟩	🟨	🟥	🟥	🟩	🟩

Figure 2: Single and Two-Object dataset design. Class labels belonging to each dataset split: train and in-distribution are highlighted in green, OOD validation in yellow, and OOD test in red.

	Train	**ID Val**	**ID Test**	**OOD Val**	**OOD Test**
Single-Object	1360	340	340	400	1100
Two-Object	7440	1860	1860	600	3700
Relational	440	110	110	250	400

Table 1: Our extended benchmark statistics for the three datasets showcasing the number of images within each of the splits.

For both ZSL and GZSL tasks, models are fine-tuned on images and labels from the training split of the data. In the ZSL setting, at test time, models must pick the correct label for an image from a set $\mathcal{S}$ of unseen labels, i.e. $\mathcal{S} \subseteq \mathcal{Y}^{\text{OODtest}}$. In the GZSL task, at test time, models must pick the correct label for an image from a set of both seen and unseen labels, i.e. $\mathcal{S} \subseteq \mathcal{Y}$ or $\mathcal{S} \subseteq \mathcal{Y}_{\text{rel}}$. This setup evaluates the ability of models to generalise colours or relations learned during fine-tuning to new unseen shape combinations. Because of this, the single and two-object train split contains at least one class containing each shape and each colour. Similarly, the relational train split contains at least one of each shape.

We only use positive examples when fine-tuning CLIP rather than both positive and negative examples to keep consistent with the DreamBooth fine-tuning method for Stable Diffusion which only accepts positive training examples. In addition, to further align with DreamBooth, we fine-tune CLIP with a small number of samples from each class (20-40 per class).

3.1 Single-Object

The single-object task tests the ability of models to recognise attribute-object pairs and is used as a baseline for analysing which combinations the models can recognise before experimenting in a two-object setting. Examples from the single-object dataset are shown in Figure 4 a) and b). In the single-object setting, we evaluate only on the GZSL task, and require models to select the correct label for the image from all possible label combinations, i.e. from the whole of $\mathcal{Y}$. Following convention, the class labels are given in the form of a prompt "a photo of a <class>".

3.2 Two-Object

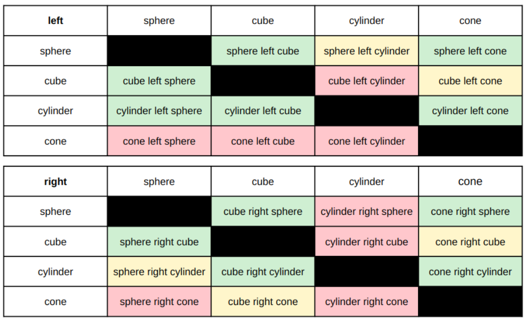
The two-object dataset contains images of exactly two-objects which differ in *both shape and colour*. For example, the dataset contains images of a *blue cube* and a *red sphere* but not of a *blue cube* and a *blue sphere*. We follow Lewis et al. (2024) and present the model with labels for individual objects whereby the true label correctly describes one of the objects in the image and the others are incorrect. In comparison to giving the model a full description of the image (e.g. *green cone and purple cylinder*), this is a challenging setup which minimises the use of shortcuts by the model, for example if the model can recognise green cones correctly but not purple cylinders. As an example, the images in Figure 4 c) may have the true label *green cylinder* and hard negatives *green cone* and *purple cylinder*.

In the ZSL setting, models are given one correct label and two distractors from the same (unseen) split. For example, an image of a yellow cube may be paired with gray cylinder and brown sphere as distractors (see Figure 4 column d)).

In the GZSL setting, models choose from five labels: the true label, two standard distractors, and two hard negatives created by swapping attributes and shapes (e.g., yellow cone, cyan cube for Figure 4 d)). This makes the task more challenging and tests whether models prefer familiar (seen) classes over novel ones.

3.3 Relational

The relational dataset tests compositions of the relations *left* and *right* between two-objects in an image. The two-objects are always two distinct shapes, that is, we don't consider cases such as *sphere left sphere*. As with the two-object dataset, each image has two possible true labels. For instance, the images in Figure 4 column e) would have the true labels *cube left sphere* and *sphere right cube*. Again, we consider a ZSL and GZSL setting. In the GZSL setting, models choose from five options: the true label, two randomly selected labels, and two hard negatives. One hard negative alters the spatial relation (e.g., cube left sphere → cube right sphere), while the other swaps object order (e.g., cube left sphere → sphere left cube). The hard negatives require the model to recognise the specific relation in the image and not just recognise which two shapes are present—a task at which a bag-of-words model would fail.



left	sphere	cube	cylinder	cone
sphere	■	sphere left cube	sphere left cylinder	sphere left cone
cube	cube left sphere	■	cube left cylinder	cube left cone
cylinder	cylinder left sphere	cylinder left cube	■	cylinder left cone
cone	cone left sphere	cone left cube	cone left cylinder	■

right	sphere	cube	cylinder	cone
sphere	■	cube right sphere	cylinder right sphere	cone right sphere
cube	sphere right cube	■	cylinder right cube	cone right cube
cylinder	sphere right cylinder	cube right cylinder	■	cone right cylinder
cone	sphere right cone	cube right cone	cylinder right cone	■

Figure 3: Relational dataset design. Class labels belonging to each dataset split: train and in-distribution are highlighted in green, OOD validation in yellow, and OOD test in red.

4 Results

We conduct experiments comparing frozen and fine-tuned CLIP, ViLT, and Diffusion Classifier (DC) on three datasets: single-object, two-object, and relational. Experiments are carried out in a Linux environment using an RTX 2080 GPU for both training and inference.

4.1 Single-Object

We test models' ability to compose single attribute-noun pairs. For each of the models we fine-tune with three different seeds and report the mean and standard deviation of each. Fine-tuning details and hyperparameters for each dataset are provided in Appendix A.

Model	ID Validation	ID Test	OOD Validation	OOD Test
Frozen CLIP	$85.29^{0.00}$	$80.59^{0.00}$	$67.75^{0.00}$	$87.36^{0.00}$
CLIP-FT	$95.29^{3.01}$	$95.59^{2.92}$	$93.57^{3.81}$	$91.21^{6.54}$
Frozen ViLT	$51.47^{0.00}$	$50.0^{0.00}$	$34.5^{0.00}$	$44.91^{0.00}$
ViLT-FT	$95.88^{0.00}$	$94.71^{0.00}$	$63.5^{0.00}$	$77.18^{0.00}$
Frozen DC	$40.80^{0.89}$	$40.98^{0.37}$	$58.0^{0.50}$	$60.0^{1.08}$
DC-FT	$97.74^{1.6}$	$97.16^{0.78}$	$99.50^{0.12}$	$99.47^{0.87}$

Table 2: Accuracy of models on the single-object task.

Results We see in Table 2 that CLIP has the best accuracy of the frozen models on this task. However, after fine-tuning, DC has the best overall accuracy. Both CLIP and DC show a strong performance on ID and OOD splits indicating that in the simple single-object setting they are able to generalise to unseen colour-shape combinations. In contrast, fine-tuned ViLT showcases strong performance only on the ID splits and shows a drop in accuracy to 63.5% and 77.18% on the OOD splits. ViLT frequently makes errors such as predicting *blue cone* for *cyan cone* or *gray cube* for *gray cylinder*—failing to generalise from familiar components seen during training (such as the colour cyan with a sphere, or the shape cylinder with other colours like red, green, or purple). Fine-tuned CLIP and DC are able to generalise in the single-object setting but ViLT's lower OOD performance shows even in simple settings composing unseen combinations can be difficult for VLMs.

4.2 Two-Object Zero-Shot

The two-object experiment tests whether models can correctly bind attributes to their corresponding objects, rather than simply recognising which shapes and colours are present. We report the average accuracy with the standard deviation for all models as shown in Table 3.

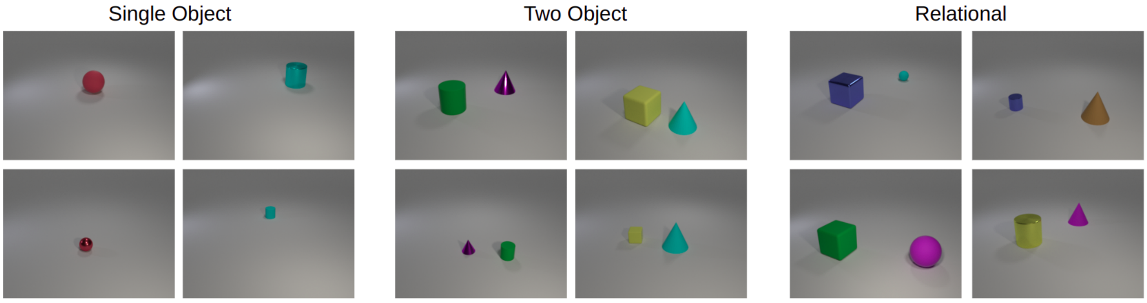
Model	ID Validation	ID Test	OOD Validation	OOD Test
Frozen CLIP	$83.71^{0.00}$	$85.27^{0.00}$	$93.0^{0.00}$	$69.51^{0.00}$
CLIP-FT	$90.13^{0.55}$	$90.39^{0.01}$	$99.39^{0.75}$	$80.15^{1.11}$
Frozen ViLT	$72.78^{0.56}$	$73.80^{0.73}$	$70.0^{0.00}$	$66.82^{0.32}$
ViLT-FT	$99.78^{0.00}$	$99.89^{0.08}$	$99.5^{0.00}$	$99.26^{0.18}$
Frozen DC	$61.18^{0.00}$	$64.53^{0.00}$	$91.83^{0.00}$	$58.3^{0.00}$
DC-FT	$82.59^{3.34}$	$83.21^{3.59}$	$93.89^{2.49}$	$72.80^{2.06}$

Table 3: Accuracy of models on the ZSL two-object task.

ViLT-FT has the highest accuracy achieving over 99% on all dataset splits. This is particularly surprising given its lower performance in the single-object task. ViLT may benefit from the reduced label space in the two-object ZSL experiment compared to having the full range of prompts in the single-object setting. CLIP-FT and DC-FT both show a decrease in performance on OOD test but not on OOD val. We believe the high OOD val accuracies are due to the reduced size of the OOD val split meaning there are only 4 very distinct prompts to choose from. The drop in performance of all models on the OOD test split further highlights that VLMs lack robust compositional understanding, even for the simpler zero-shot case. Current pre-training strategies rarely require models to explicitly learn compositional knowledge, suggesting that adjustments to pre-training may be necessary.

4.3 Two-Object Generalised Zero-Shot

In the GZSL two-object task, models must compose attributes with objects while also handling previously seen labels, providing a more rigorous test of generalisability. We report the accuracies



a) red sphere b) cyan cube c) green cylinder d) yellow cube e) sphere right cube f) cylinder left cone

Figure 4: Samples from our extended benchmark with two example classes displayed from each dataset—single, two-object, and relational.

and standard deviations for the two-object GZSL experiment in Table 4.

Model	ID Validation	ID Test	OOD Validation	OOD Test
Frozen CLIP	$23.33^{0.00}$	$21.56^{0.00}$	$35.33^{0.00}$	$34.27^{0.00}$
CLIP-FT	$78.82^{3.05}$	$76.40^{0.86}$	$55.50^{5.92}$	$23.38^{5.28}$
Frozen ViLT	$31.56^{0.12}$	$32.71^{0.29}$	$47.83^{0.00}$	$29.1^{0.21}$
ViLT-FT	$99.71^{0.07}$	$99.86^{0.03}$	$91.67^{0.00}$	$83.46^{0.06}$
Frozen DC	$33.58^{0.00}$	$34.64^{0.00}$	$38.46^{0.00}$	$39.32^{0.00}$
DC-FT	$53.06^{3.20}$	$51.86^{3.41}$	$57.06^{5.03}$	$72.97^{2.05}$

Table 4: Accuracy of models on GZSL two-object.

Again ViLT-FT has the strongest performance for all dataset splits significantly outperforming other models. This suggests it is less biased towards seen labels as evidenced by the relatively stable performance across ZSL and GZSL. However, it does still exhibit a small drop in performance on the OOD splits indicating some limitations in generalising. CLIP-FT experiences a substantial drop in performance on the OOD splits especially OOD test, showing it has overfit to the training data. DC-FT interestingly shows the reverse pattern to the other models and has the highest accuracy on OOD. We hypothesise that this is due to the composition of the test split—for example, challenging colours like yellow and brown, which DC frequently confuses, constitute a smaller proportion of the OOD labels. While the high OOD test accuracy is particularly notable in the challenging GZSL setting, DC's lower accuracy on the ID splits (53.06% and 51.86%) suggests it lacks consistent attribute-object binding ability. Even ViLT-FT, the best-performing model overall, has a reduced performance on the OOD splits, highlighting limitations in the way models represent and combine attributes and objects.

In table 5 we show the percentage of total predictions made by the models which fall into each error category on the GZSL two-object task for the ID and OOD test splits. The *Colour* column is the percentage of predictions where the model correctly identifies the shape but chooses the colour of the second object in the image, the *Shape* column is the percentage of predictions correctly guessing the colour but choosing the second object's shape. The *Other* column is the predictions from the other two non-hard negatives.

Model	ID test			OOD Test		
	Colour	Shape	Other	Colour	Shape	Other
Frozen CLIP	$35.97^{0.0}$	$37.42^{0.0}$	$5.06^{0.0}$	$22.05^{0.0}$	$30.03^{0.0}$	$13.65^{0.0}$
CLIP-FT	$7.15^{1.58}$	$8.30^{0.53}$	$8.16^{1.28}$	$34.17^{3.64}$	$34.78^{2.95}$	$7.72^{1.29}$
Frozen ViLT	$35.57^{0.4}$	$26.54^{0.47}$	$5.18^{0.51}$	$35.98^{0.05}$	$28.56^{0.19}$	$6.35^{0.02}$
ViLT-FT	$0.03^{0.04}$	$0.0^{0.0}$	$0.0^{0.0}$	$8.76^{0.02}$	$7.54^{0.02}$	$0.17^{0.04}$
Frozen DC	$21.46^{0.0}$	$26.25^{0.0}$	$17.64^{0.0}$	$14.70^{0.0}$	$18.18^{0.0}$	$27.81^{0.0}$
DC-FT	$18.53^{0.55}$	$16.79^{1.24}$	$12.81^{1.63}$	$16.51^{0.71}$	$10.53^{1.75}$	$0.0^{0.0}$

Table 5: Breakdown of errors in two-object GZSL.

Both frozen and fine-tuned CLIP have a roughly even distribution of errors on colour and shape mistakes showing both types of composition are equally challenging. Frozen ViLT makes slightly more errors on colour, but after fine-tuning, errors across all categories drop to near zero with a slight tendency for colour errors remaining in OOD. Frozen DC makes slightly more mistakes on shape but after fine-tuning finds colour slightly more difficult especially in OOD. All incorrect predictions made by DC-FT in the OOD split correspond to hard negative labels, highlighting that binding colours to the correct objects is particularly challenging.

4.4 Relational Zero-Shot

The relational experiment tests how well models can compose spatial relations with objects, specifically we test the composition of the relations 'left' and 'right' with the object's shape eg. 'cube'. We show the relational ZSL results in Table 6.

Model	ID Validation	ID Test	OOD Validation	OOD Test
Frozen CLIP	$56.36^{0.00}$	$56.60^{0.00}$	$38.40^{0.00}$	$68.00^{0.00}$
CLIP-FT	$99.39^{0.86}$	$99.31^{0.57}$	$68.00^{13.91}$	$94.08^{3.86}$
Frozen ViLT	$74.55^{1.48}$	$68.52^{0.87}$	$42.40^{0.00}$	$64.67^{0.31}$
ViLT-FT	$78.18^{2.57}$	$76.04^{1.98}$	$70.53^{0.19}$	$65.0^{0.35}$
Frozen DC	$68.18^{0.00}$	$69.44^{0.00}$	$30.70^{0.00}$	$65.25^{0.00}$
DC-FT	$89.09^{4.64}$	$92.94^{1.18}$	$51.86^{2.31}$	$87.18^{9.18}$

Table 6: Accuracy of models on the ZSL relational task.

All models except ViLT-FT have a lower accuracy on OOD validation than OOD test. This could be due to the smaller size of the validation split, which limits prompt diversity making the distractor labels more likely to share shapes with the shapes in the true label. Both DC and CLIP only show slight drops in performance between OOD test and the ID splits demonstrating the capacity to recognise unseen object-relation combinations in ZSL settings. ViLT, while having overall lower accuracies, shows less variation across dataset splits, showing some capacity to generalise. All models show a substantial drop in performance in the relational ZSL compared with the two-object ZSL showing that systematically combining objects with relations is harder for these models than combining colour-object pairs. The difficulty the models have with relational information suggests they are focusing on recognising objects in the image rather than compositions between objects. While VLMs can often rely on these shortcuts and still achieve a strong performance, tasks that require relational reasoning reveal that they lack a full understanding of visual scenes.

4.5 Relational Generalised Zero-Shot

In the relational GZSL experiment, models must bind spatial relations to objects and predict previously unseen combinations of relations and objects in the presence of previously seen labels. We show the performance of the models on the GZSL relational task in Table 7 reporting the mean and standard deviation for the fine-tuned models.

Model	ID Validation	ID Test	OOD Validation	OOD Test
Frozen CLIP	$27.27^{0.00}$	$27.43^{0.00}$	$18.00^{0.00}$	$25.0^{0.00}$
CLIP-FT	$62.12^{0.43}$	$72.22^{3.97}$	$42.80^{18.39}$	$34.75^{16.33}$
Frozen ViLT	$13.94^{0.43}$	$16.55^{0.65}$	$22.53^{0.19}$	$26.5^{0.35}$
ViLT-FT	$16.55^{0.65}$	$22.53^{0.19}$	$26.5^{0.35}$	$25.50^{1.08}$
Frozen DC	$24.55^{0.00}$	$21.53^{0.00}$	$10.00^{0.00}$	$24.50^{0.00}$
DC-FT	$32.73^{2.57}$	$34.72^{2.60}$	$41.20^{4.57}$	$38.25^{4.02}$

Table 7: Accuracy of models on relational GZSL.

In the GZSL relational setting, CLIP-FT performs reasonably well on the ID splits with 62.12% and 72.22%, however, there is a significant drop in performance for the OOD splits with 42.80% and 34.75% on validation and test respectively. CLIP therefore seems to overfit to the training data and is not able to generalise to unseen labels. ViLT struggles with this task, with even the fine-tuned model hardly performing better than chance at 20%. Interestingly, DC has a lower accuracy on the ID splits than the OOD splits. Given DC's reasonable accuracies of 89.09% and 92.94% in the ID ZSL experiment, it appears DC is particularly confused by the presence of hard negative labels showing it is lacking fine-grained understanding. All models have a drop in performance from the ZSL task showing they struggle to compose relational con-

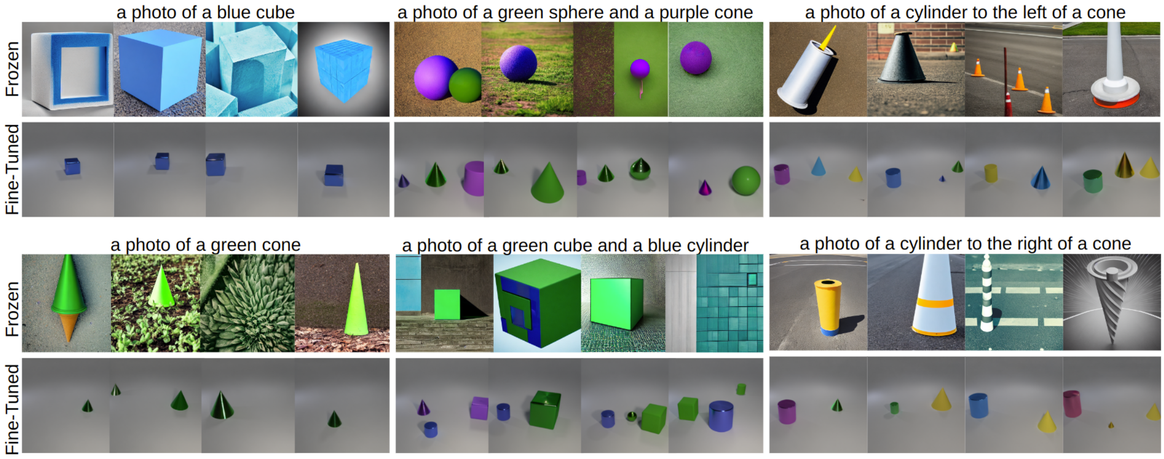

a photo of a blue cube
a photo of a green sphere and a purple cone
a photo of a cylinder to the left of a cone
Frozen
Fine-Tuned
a photo of a green cone
a photo of a green cube and a blue cylinder
a photo of a cylinder to the right of a cone
Frozen
Fine-Tuned

Figure 5: Images generated by Frozen and Fine-Tuned Diffusion-Classifier using prompts from the single, two-object and relational, shown from left to right. The top two rows are generated by labels from the train set and the bottom two from the test set.

cepts and especially cannot distinguish between hard negatives in the GZSL setting such as *sphere left cube* and *sphere right cube*. This suggests that the models are relying on object recognition rather than understanding relational positions.

We show the percentage of total predictions made by the models which fall into each error category for the ID and OOD test splits in Table 8. The column Left/Right shows the percentage of predictions which choose the hard negative where only the relation is incorrect e.g. *cube left sphere* instead of *cube right sphere*. The Shape column displays the percentage of predictions where the shapes are correct but in the incorrect order e.g. *sphere left cube* instead of *cube left sphere*. The Other column is the predictions from the other two non-hard negative labels.

Model	**ID test**			**OOD Test**		
	Left/Right	Shape	Other	Left/Right	Shape	Other
Frozen CLIP	$26.04^{0.0}$	$28.13^{0.0}$	$18.41^{0.0}$	$20.0^{0.0}$	$25.5^{0.0}$	$29.5^{0.0}$
CLIP-FT	$10.07^{1.99}$	$17.59^{3.46}$	$0.12^{0.16}$	$30.50^{10.5}$	$29.0^{5.75}$	$5.75^{4.13}$
Frozen ViLT	$31.25^{0.75}$	$36.34^{0.65}$	$15.86^{1.28}$	$8.0^{0.2}$	$40.67^{1.0}$	$24.83^{1.05}$
ViLT-FT	$25.87^{0.19}$	$24.40^{0.75}$	$27.20^{0.33}$	$9.08^{1.0}$	$36.42^{1.04}$	$28.83^{0.72}$
Frozen DC	$25.0^{0.0}$	$30.56^{0.0}$	$22.92^{0.0}$	$22.25^{0.0}$	$22.5^{0.0}$	$30.75^{0.0}$
DC-FT	$24.08^{1.99}$	$36.92^{1.34}$	$4.28^{2.78}$	$22.33^{1.48}$	$28.92^{5.60}$	$10.5^{4.30}$

Table 8: Breakdown of errors in relational GZSL.

Frozen CLIP and DC have fairly evenly distributed errors across the 3 categories while ViLT on the OOD test split has a larger proportion of shape errors. After fine-tuning, DC and CLIP have a reduced proportion of errors in the other category however ViLT still makes a considerable number of predictions where the two shapes in the image are not correctly identified. For fine-tuned CLIP and DC, the hard negative captions present the most difficulty with both types of hard negative being frequently predicted instead of the true label.

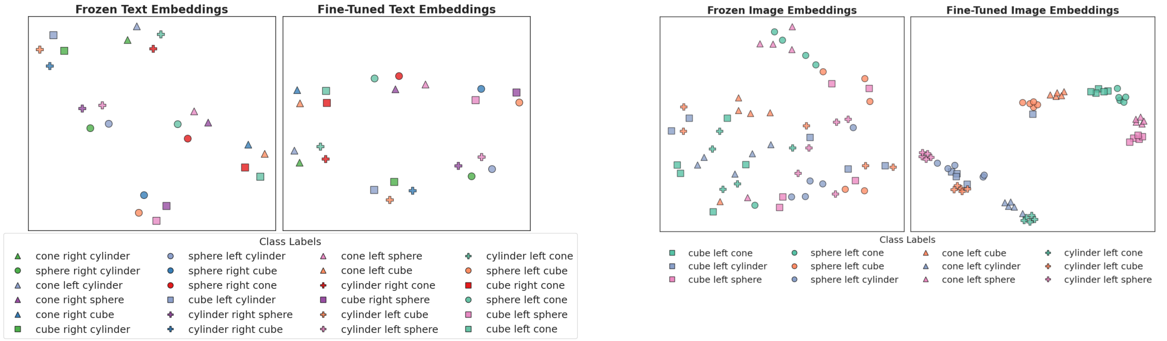
5 Model Understanding

Stable Diffusion Images We compare images generated by frozen and fine-tuned Stable Diffusion to evaluate what features Diffusion Classifier is able to learn from fine-tuning on each dataset. We use a guidance scale of 7 and 50 inference steps. Examples using prompts from each dataset are shown in Figure 5. Frozen Stable Diffusion is generally very poor at generating images in alignment with the specified prompt, except in the single-object case. Interestingly, the two-object and relational fine-tuned Stable Diffusion generate three objects fairly frequently showing some pre-training bias and knowledge is still preserved. The relational fine-tuned model fails to understand the difference between the left and right relations with the prompts “a cylinder to the left of a cone” and “a cylinder to the right of a cone” both resulting in images of a cylinder on the left—the class seen during training.

CLIP embeddings We show t-SNE visualisations of image and text embeddings from relational dataset examples for frozen and fine-tuned CLIP. For images, we show the embeddings of 5 samples from each class and only consider classes containing *left* since corresponding classes containing *right* use the same images.

Figure 6a shows the text embeddings which are clearly clustered into quadruples corresponding to prompts where the object shapes are the same, with no clear separation between prompts corresponding to different arrangements of objects. For example, the closest neighbours of *cube left sphere* are *sphere left cube*, *cube right sphere* and *sphere right cube*. Fine-tuning (right-hand plot) fails in most cases to overcome this clustering of similar prompts. An exception is the cluster of prompts containing *sphere left cube* and *cube right sphere*, which have been moved closer together, and are visibly distinct from *sphere right cube* and *cube left sphere*. Other groups of prompts tend to cluster according to ordering of nouns (e.g. *cube left cone* and *cube right cone*), or by bag-of-words similarity (e.g. *cube left cylinder* and *cylinder left cube*). This inability to distinguish prompts corresponding to different arrangements of objects likely contributes towards CLIP’s inability to correctly caption images with the same shapes but different relations.

The t-SNE visualisation of image embeddings presented in Figure 6b shows that images belonging to the same class are mostly well-clustered. However, there are a few instances of classes in the wrong cluster e.g. a *cube left cylinder* sample appears within the *sphere left cube* cluster. Notably, we observe that embeddings of images with reversed relational structures tend to occupy similar regions in the space—for instance, *cylinder left cone* and *cone left cylinder* appear close together at the bottom of the plot, while *cylinder left sphere* and *sphere left cylinder* are both near the left side of the plot. This spatial overlap may contribute to CLIP’s difficulty with relational reasoning. OOD classes such as *cube left cylinder* and *sphere left cylinder*, which are not directly fine-tuned, appear slightly less well clustered.



(a) t-SNE visualisation of frozen and fine-tuned CLIP text embeddings for relational prompts. Best viewed electronically or in colour.

(b) t-SNE visualisation of frozen and fine-tuned CLIP image embeddings for relational prompts. Best viewed electronically or in colour.

Figure 6: t-SNE visualisations of CLIP text and image embeddings for relational prompts after fine-tuning.

6 Discussion

We extend the Concept Binding Benchmark from Lewis et al. (2024) to assess concept binding in zero-shot (ZSL) and generalised zero-shot (GZSL) settings. Using this extended framework, we compare the performance of the discriminative models CLIP and ViLT against a generative model, Diffusion Classifier, on single-object, two-object, and relational compositional tasks. Diffusion Classifier shows the highest generalisation accuracy on the single-object task. ViLT achieves state-of-the-art performance on both ZSL and GZSL two-object tasks, demonstrating strong compositional ability in binding attributes to objects even in GZSL settings. Diffusion Classifier shows some capacity to generalise in the two-object GZSL setting, however, it falls short of ViLT's performance.

On the relational composition task, all models perform poorly, showing considerable drops in performance on the GZSL from the ZSL task showing that hard distractors such as *cube left sphere* versus *cube right sphere* are a particular problem. Despite initial hopes that Diffusion Classifier's generative approach might better handle compositionality, relational reasoning remains a major challenge for all models tested.

On all our experiments, our fine-tuned CLIP model consistently outperforms the model from Lewis et al. on the OOD splits (Lewis et al., 2024). We attribute this to our fine-tuning strategy of only using positive examples unlike Lewis et al. who use both positive and negative examples. We hypothesise that the inclusion of negative examples exacerbates overfitting. This is due to prompts appearing as negative training examples which then appear as positive examples in the OOD splits, causing CLIP to suppress their prediction. Therefore our positive-only approach appears to lead to better generalisation and reduced overfitting.

The low performance on the GZSL relational task suggests current VLMs may rely too heavily on shortcuts such as object recognition rather than developing structured, compositional representations. Our analysis of image and text embeddings in CLIP further supports this: relational concepts (e.g., *left* vs. *right*) are not sufficiently disentangled, especially in the text embedding space, limiting the models' capacity to reason about spatial relationships. Potential avenues to address this are training on datasets with more explicit compositional objectives and developing better prompting or fine-tuning strategies that encourage attribute and relation disentanglement. Further work in this area also includes analysis of where exactly the models fail: do they fail in forming correct representations of individual words, or do they fail in how these words are combined? This could be tackled using a formal semantic approach that has been integrated with a vector-based semantics, such as in Coecke et al. (2010) or Baroni and Zamparelli (2010).

However, while these routes to improved compositional understanding are important, we argue that our results highlight an important limitation of the tested models as they stand: at present compositional understanding is clearly limited. Since there may be a number of aspects of composition that we require models to perform, these should be considered at the pre-training stage rather than expecting users to fine-tune for these fundamental semantic abilities.

Limitations

While our benchmark uses synthetic, simplistic images, we chose this design specifically to reduce the risk of spurious correlations (Wu et al., 2023) and enable precise compositional structures to be tested for. We view this benchmark as a diagnostic test for probing specific compositional generalisation properties in VLMs that may be masked in more complex, real-world scenarios. Future work could include expanding these experiments to test other attributes such as material or size. Another interesting avenue for future research would be to expand the experiments to include more than two objects.

Acknowledgements

The authors wish to acknowledge and thank the financial support of the UK Research and Innovation (UKRI) [Grant ref EP/S022937/1] and the University of Bristol. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol—http://www.bristol.ac.uk/acrc/

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1183–1193.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and 1 others. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402.

Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gul Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, and 1 others. 2023. Going beyond nouns with vision & language models using synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20155–20165.

Huanran Chen, Yinpeng Dong, Zhengyi Wang, X. Yang, Chen-Dong Duan, Hang Su, and Jun Zhu. 2023. Robust classification via a single diffusion model. *ArXiv*, abs/2305.15241.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton and Co., The Hague.

Kevin Clark and Priyank Jaini. 2023. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen J Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36(1):345–384.

Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. 2022. Why is winoground hard? investigating failures in visuolinguistic compositionality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. 2023. Teaching structured vision & language concepts to vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2668.

Sri Harsha Dumpala, Aman Jaiswal, Chandramouli Sastry, Evangelos Milios, Sageev Oore, and Hassan Sajjad. 2024. Sugarcrepe++ dataset: Vision-language model sensitivity to semantic and lexical alterations. *arXiv preprint arXiv:2406.11171*.

Xuehai He, Weixi Feng, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. 2023. Discriminative diffusion models as few-shot vision and language learners. *arXiv preprint arXiv:2305.10722*.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2024. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in Neural Information Processing Systems*, 36.

Hang Hua, Jing Shi, Kushal Kafle, Simon Jenni, Daoan Zhang, John Collomosse, Scott Cohen, and Jiebo Luo. 2024a. Finematch: Aspect-based fine-grained image and text mismatch detection and correction. In *European Conference on Computer Vision*, pages 474–491. Springer.

Hang Hua, Yunlong Tang, Ziyun Zeng, Liangliang Cao, Zhengyuan Yang, Hangfeng He, Chenliang Xu, and Jiebo Luo. 2024b. Mmcomposition: Revisiting the compositionality of pre-trained vision-language models. *arXiv preprint arXiv:2410.09733*.

Irene Huang, Wei Lin, Muhammad Jehanzeb Mirza, Jacob Hansen, Sivan Doveh, Victor Butoi, Roei Herzig, Assaf Arbelle, Hilde Kuehne, Trevor Darrell, Chuang Gan, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. 2024. Conme: Rethinking evaluation of compositional reasoning for modern vlms. In *Advances in Neural Information Processing Systems*, volume 37, pages 22927–22946. Curran Associates, Inc.

Theo MV Janssen and Barbara H Partee. 1997. Compositionality. In *Handbook of logic and language*, pages 417–473. Elsevier.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR.

Benno Krojer, Elinor Poole-Dayan, Vikram Voleti, Christopher Pal, and Siva Reddy. 2023. Are diffusion models vision-and-language reasoners? In *NeurIPS*.

Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. 2024. Does CLIP bind concepts? probing compositionality in large image models. In *Findings EACL 2024*, pages 1487–1500, St. Julian's, Malta. Association for Computational Linguistics.

Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. 2023. Your diffusion model is secretly a zero-shot classifier. In *ICCV*.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2023. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10910–10921.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language: Proceedings of the 1970 Stanford workshop on grammar and semantics*, pages 221–242. Springer.

Rohan Pandey, Rulin Shao, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2023. Cross-modal attention congruence regularization for vision-language relation alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5444–5455, Toronto, Canada. Association for Computational Linguistics.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. 2022. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4051–4070.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan Plummer, Ranjay Krishna, and Kate Saenko. 2024. Cola: A benchmark for compositional text-to-image retrieval. *Advances in Neural Information Processing Systems*, 36.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Ugur Sahin, Hang Li, Qadeer Khan, Daniel Cremers, and Volker Tresp. 2024. Enhancing multimodal compositional reasoning of visual language models with generative negative mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5563–5573.

Ziyi Shou and Fangzhen Lin. 2024. Enhancing semantic understanding in vision language models using meaning representation negative generation. In *Fourth Workshop on Knowledge-infused Learning*.

Harman Singh, Pengchuan Zhang, Qifan Wang, Mengjiao Wang, Wenhan Xiong, Jingfei Du, and Yu Chen. 2023. Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 869–893, Singapore. Association for Computational Linguistics.

Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. 2024. A survey on compositional learning of AI models: Theoretical and experimental practices. *Transactions on Machine Learning Research*. Survey Certification.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248.

Ankit Vani, Bac Nguyen, Samuel Lavoie, Ranjay Krishna, and Aaron Courville. 2024. Sparo: Selective attention for robust and compositional transformer encodings for vision. In *European Conference on Computer Vision*, pages 233–251. Springer.

Chenwei Wu, Li Erran Li, Stefano Ermon, Patrick Haffner, Rong Ge, and Zaiwei Zhang. 2023. The role of linguistic priors in measuring compositional generalization of vision-language models. In *Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops*, volume 239 of *Proceedings of Machine Learning Research*, pages 118–126. PMLR.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591.

Nir Yellinek, Leonid Karlinsky, and Raja Giryes. 2025. 3vl: Using trees to improve vision-language models' interpretability. *Trans. Img. Proc.*, 34:495–509.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*.

Le Zhang, Rabiul Awal, and Aishwarya Agrawal. 2024. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13774–13784.

Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. 2022. An explainable toolbox for evaluating pretrained vision-language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 30–37.

A Fine-tuning Details

Optimal hyper-parameters were selected by performing a search for each model. We consider the parameters: learning rate, images per class, epochs and LoRA parameters where applicable. We select final parameters based on averaged performance on the ID val and OOD val dataset splits.

Single-Object CLIP was fine-tuned using 40 images per class for 30 epochs, using an Adam optimiser with a learning rate of $1 \cdot 10^{-6}$, a batch size of 16, and a contrastive loss. For DC, we used DreamBooth to fine-tune Stable Diffusion's U-Net and text-encoder. We use 30 images per class for 4000 steps with a learning rate of $5 \cdot 10^{-6}$ and a batch size of 1. All inferences were performed using 200 noise samples. ViLT was fine-tuned on 80 images per class using LoRA with a learning rate of $1 \cdot 10^{-5}$ setting the LoRA rank (r) to 16 and the scaling factor (alpha) to 32.

Two-Object CLIP was fine-tuned using 40 images per class for 30 epochs, using an Adam optimiser with a learning rate of $1 \cdot 10^{-6}$, a batch size of 16, and a contrastive loss. For DC, we fine-tuned using 30 images per class for 4000 steps with a learning rate of $5 \cdot 10^{-6}$ and a batch size of 1. All inferences were performed using 200 noise samples. ViLT was fine-tuned using LoRA with a learning rate of $1 \cdot 10^{-5}$ setting the LoRA rank (r) to 16 and the scaling factor (alpha) to 32.

Relational CLIP uses the same parameters as the single-object model except using 20 images per class for 50 epochs. DC is fine-tuned on 40 images per class for 5000 steps with the remaining parameters the same as the previous two models. ViLT is fine-tuned on 40 images per class using LoRA with a learning rate of $1 \cdot 10^{-6}$ setting the LoRA rank (r) to 8 and the scaling factor (alpha) to 16.

On the Distinctive Co-occurrence Characteristics of Antonymy

Zhihan Cao
Institute of Science Tokyo
School of Computing
`cao.z.c8a7@m.isct.ac.jp`

Hiroaki Yamada
Institute of Science Tokyo
School of Computing
`yamada@comp.isct.ac.jp`

Takenobu Tokunaga
Institute of Science Tokyo
School of Computing
`take@c.titech.ac.jp`

Abstract

Antonymy has long received particular attention in lexical semantics. Previous studies have shown that antonym pairs frequently co-occur in text, across genres and parts of speech, more often than would be expected by chance. However, whether this co-occurrence pattern is distinctive of antonymy remains unclear, due to a lack of comparison with other semantic relations. This work fills the gap by comparing antonymy with three other relations across parts of speech using robust co-occurrence metrics. We find that antonymy is distinctive in three respects: antonym pairs co-occur with high strength, in a preferred linear order, and within short spans. All results are available online.

1 Introduction

Among various semantic relations, antonymy has long received particular attention. Charles and Miller (1989) proposed *the co-occurrence hypothesis*, suggesting that adjectival antonymy pairs tend to co-occur within a sentence more often than expected by chance. Subsequent studies have provided empirical support for this hypothesis (Justeson and Katz, 1991). This co-occurrence tendency has also been observed in parts of speech (PoS) other than adjectives (Fellbaum, 1995; Mohammad et al., 2008) and across various genres (Jones and Murphy, 2005; Jones, 2006).

Prior studies have identified several characteristics of the co-occurrence of antonymy pairs. One is that co-occurrence is likely to be ordered. Each antonymy pair often appears in a specific linear order (Kostić, 2015, 2017; Wu and Zhang, 2022). The more frequent antonym tends to precede its less frequent counterpart (Jones, 2002). Another characteristic is that antonymy pairs likely co-occur in close proximity. This proximity typically falls within five-word windows or appears in coordinated constructions (Jones et al., 2007; Mohammad et al., 2008). Therefore, antonymy pairs are characterised by an intra-sentential co-occurrence pattern, which might be distinctive of antonymy.

However, this assumption remains largely untested, as the intra-sentential co-occurrence characteristics of non-antonymy relations have been underexplored. Some indirect evidence does exist. For example, hypernymy or holonymy pairs can be retrieved using lexico-syntactic patterns (Hearst, 1992; Girju et al., 2006; Roller et al., 2018). Yet these approaches rely on specific constructions rather than reflecting the general co-occurrence patterns within sentences. As a result, existing work is insufficient to determine whether the co-occurrence pattern uniquely characterises antonymy.

There is also a methodological concern. Prior studies commonly relied on raw co-occurrence counts or pointwise mutual information (PMI) of word pairs. Such metrics are biased toward extremely low or high frequencies that are common due to the sparsity of word frequencies (Schulte Im Walde and Melinger, 2008). The sparsity also makes many statistical tests (such as the Chi-square test used by Kostić (2017)) unreliable (Dunning, 1993), raising concerns about whether the co-occurrence observed is truly significant.

If antonymy indeed stands out in its co-occurrence, this distinctiveness needs to be established rather than assumed. Understanding different semantic relations in natural text is foundational not only to linguistic theory but also to computational modeling. Recently, pretrained language models (PLMs) have exhibited a puzzling phenomenon, *antonymy bias*. Regardless of architecture and size, PLMs perform substantially better on predicting the antonym of a given word over other relations (Pitarch et al., 2023; Cao et al., 2025a). Establishing the distinctive co-occurrence characteristics of antonymy will lay the groundwork for analysing the semantic behavior of PLMs trained on a plain corpus.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 134–141
November 8-9, 2025 ©2025 Association for Computational Linguistics

Research Objective In summary, although prior studies have suggested that antonymy exhibits distinctive co-occurrence characteristics, existing evidence remains insufficient to validate this assumption. The present work evaluates and compares the co-occurrence of antonymy and other semantic relations across PoS, using robust and statistically grounded co-occurrence metrics. We aim to establish whether antonymy presents a unique pattern of co-occurrence characteristics.

2 Metrics

Semantic relations theoretically hold between word lemmas that share the PoS. We then define metrics at the lemma level.

2.1 Strength of Co-occurrence

The first metric, G^2 score (Dunning, 1993), measures the co-occurrence strength. It directly addresses the methodological concerns of prior studies because it is theoretically and empirically robust to data sparsity (Moore, 2004; Evert, 2009) and supports a reliable statistical test.

The G^2 score of a word pair (w, v) is essentially the logarithm of the likelihood ratio test statistic. The underlying null hypothesis is that (w, v) co-occurs in the same sentence at a chance level[1]. G^2 scores asymptotically follow a χ^2 distribution with one degree of freedom. This allows us to interpret the score in two ways: 1) whether the co-occurrence is statistically significant, and 2) if it is, a higher score means stronger co-occurrence.

Formally, the G^2 score is defined as

$$G^2(w, v) = 2 \sum_{c \in C} O_c \log \frac{O_c}{E_c}, \tag{1}$$

where C denotes a set of the possible events regarding the co-occurrence of (w, v) in a sentence: $\{wv, w\bar{v}, \bar{w}v, \bar{w}\bar{v}\}$. Here, wv indicates that both w and v appear in the same sentence; $w\bar{v}$ and $\bar{w}v$ denote that only w or only v appears, respectively; $\bar{w}\bar{v}$ denotes that neither occurs. O_c and E_c represent the observed and expected number of sentences for each event c. The expected count E_c is calculated under the assumption that w and v occur independently. For example, when $c = wv$, the expected count is computed as $E_c = \frac{|w| \times |v|}{N}$ where $|w|$ and $|v|$ are the number of sentences containing w and v, respectively, and N is the total sentence number.

The advantages of G^2 scores are three-fold. First, G^2 scores normalise the co-occurrence of two words by their individual frequency and are hence robust to extremely high frequencies. Second, the use of logarithm mitigates the effect of extremely low frequencies. Finally, G^2 scores incorporate more information than PMI and raw counts by considering both how often a pair co-occurs and how often it does not. This enables an evaluation of the co-occurrence strength more comprehensively without being biased only by the co-occurred side.

2.2 Order of Co-occurrence

Prior studies (Kostić, 2015, etc.) suggest the order might distinguish antonymy from other relations. Our second metric evaluates the linear order of a word pair (w, v). We let w always be the more frequent word and v the less frequent one. For each sentence where (w, v) co-occurs, we assign an order score of $+1$ if w precedes v, and -1 otherwise. Then, for every word pair (w, v), we obtain a list of order scores. We determine whether there is a preferred order of (w, v) by conducting a binomial test on whether the proportion of $+1$ is significantly greater or less than 0.5. The order of a pair is defined as the average order score if the pair has a preferred order, and zero otherwise.

2.3 Distance of Co-occurrence

Whether the co-occurrence takes place in close proximity may be a distinctive characteristic of antonymy (Mohammad et al., 2008, etc.). The last metric measures the distance of co-occurrence, defined by the average number of words separating the two co-occurring words in a sentence.

Statistical Tests In order to establish whether the antonymy is statistically different from other relations, for each metric, we conduct a Brunner-Munzel test on every possible relation pair. It generalises the Mann-Whitney U test by relaxing the assumption of equal variances between samples being compared, showing better empirical robustness (Brunner and Munzel, 2000). For statistical tests described above (test of individual G^2 scores, binomial test, and Brunner-Munzel test), the significance level is set at 0.01.

pos relation NOUN HYP-HPO 11664

---

[1] The original null hypothesis is that (w, v) co-occurs adjacently by chance. We relax it according to our research objective.

POS	ANT	HOL	HYP	SYN	UNR
NOUN	97	758	11,664	1,078	7,618
VERB	74	–	6,853	1,214	869
ADJ	192	–	–	309	1,409
ADV	20	–	–	140	104
Total	383	758	18,517	2,741	10,000

Table 1: Counts of lemma pairs. "–" means the relation is not defined for the PoS in WordNet.

3 Data

We evaluate four semantic relations: antonymy (ANT), synonymy (SYN), hyper-hyponymy (HYP), and holo-meronymy (HOL). The hyper-hyponymy is defined to hold if either word in a pair is a hypernym of the other; similarly, holo-meronymy holds if either is a holonym of the other. This treatment aligns the definitions of all relations with the symmetric nature of antonymy and synonymy, making all relations comparable.

We retrieve lemma pairs of nouns, verbs, adjectives, and adverbs in any of the four relations from WordNet (Miller, 1995). A series of filtering steps is applied in order to ensure that all lemma pairs are of the same linguistic unit level and are lexico-semantically interpretable.

We exclude: 1) pairs containing multi-word expressions, abbreviations, or named entities; 2) pairs where either word has a frequency of zero or one in WordNet; 3) pairs where multiple semantic relations exist between the two lemmas; 4) verb pairs involving linking verbs, auxiliary verbs, or light verbs; 5) hypernymy pairs with a path length more than two in WordNet hierarchy[2]. These procedures result in 25,115 lemma pairs.

We estimate the metrics on the Corpus of Contemporary American English (COCA, Davies, 2008). It is designed to reflect the general usage patterns of the English language across balanced genres.

Choosing COCA brings an additional benefit: it is annotated with both PoS tags and lemmas. The PoS and lemma annotations allow us to efficiently check semantic relations between two co-occurring lemmas using WordNet.

To reduce noise, we exclude sentences shorter than five words, which are often exclamatory or fragmentary. This leaves 17,718,403 sentences for analysis. Among the 25,115 lemma pairs retrieved from WordNet, 22,399 lemma pairs are observed in COCA.

As a control, we additionally randomly sample 10,000 lemma pairs from all lemma pairs that co-occur intra-sententially but are unrelated (UNR) in any relation defined in WordNet. The filtering procedures described above are applied to unrelated pairs as well.

We sort the two lemmas in each pair by descending frequency. Table 1 shows statistics of lemma pairs for each relation. The counts are highly imbalanced across relations, reflecting the sparsity of WordNet (Cao et al., 2025b).

PoS	ANT	HOL	HYP	SYN	UNR
NOUN	**11,144**	1,819	406	628	19
	99%	92%	75%	88%	36%
VERB	**915**	–	83	164	7
	91%	–	52%	72%	24%
ADJ	**2,309**	–	–	120	9
	97%	–	–	76%	36%
ADV	**1,471**	–	–	128	14
	95%	–	–	71%	33%

Table 2: Average G^2 scores (above) and percentages of significant cases (below).

4 Results

Table 2 presents the average G^2 scores and the percentage of pairs with significant scores. In this table and all subsequent tables, boldface indicates relations whose score differs significantly from all other relations within the same PoS for the corresponding metric. Antonymy pairs consistently yield both the highest G^2 scores (ranging from 915 to 11,144) and the largest percentage (at least 91%) of significant co-occurring pairs across all PoS. Figure 1 shows the distribution of G^2 scores per relation and PoS. The first and third quantiles of G^2 scores for Antonymy are higher than those of all other relations across PoS. These results confirm that antonyms not only co-occur more frequently than expected by chance, but also with greater strength than other relations.

We acknowledge that antonymy tends to have extremely high G^2 scores. For nominal antonymy, there are even two pairs, ("child","parent") and ("man","woman"), whose G^2 scores are greater than 100,000. Such extremely high outliers might have inflated the average G^2 scores to 11,144 for nominal antonymy, making it the highest among all PoS and relations.

---

[2] Because hypernymy pairs with a path length of two or less are most intuitive to human judgments (Vulić et al., 2017).

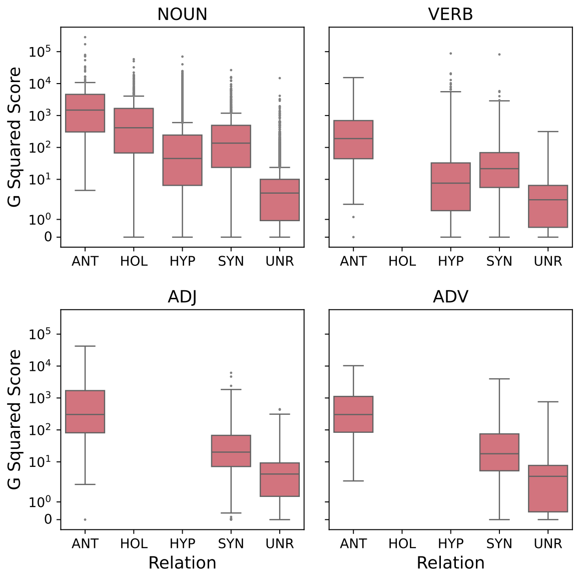


Figure 1: Distribution of G^2 scores per PoS and relation with the y axis in logarithmic scale.

PoS	ANT	HOL	HYP	SYN	UNR
NOUN	**.13** 97%	.05 96%	.02 93%	.02 90%	.01 33%
VERB	**.11** 94%	– –	.03 88%	.02 81%	.02 40%
ADJ	**.11** 91%	– –	– –	.04 61%	.00 21%
ADV	**.29** 95%	– –	– –	.05 87%	.03 29%

Table 3: Average order scores (above) and the percentages of pairs with a preferred order among all significantly co-occurring pairs (below).

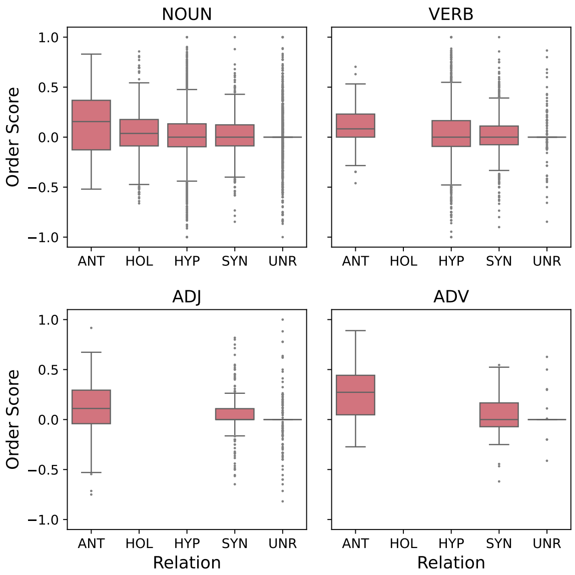


Figure 2: Distribution of order scores per PoS and relation.

PoS	ANT	HOL	HYP	SYN	UNR
NOUN	**15**	22	26	25	30
VERB	**18**	–	28	25	50
ADJ	**14**	–	–	17	25
ADV	**11**	–	–	20	27

Table 4: Average distances between significantly co-occurring pairs.

Table 3 presents the results of the order preference. Antonymy pairs show a strong tendency to prefer a specific order. Across all PoS, more than 90% of the co-occurring antonymy pairs have a preferred order, typically with an average order score above 0.10. The average order scores for antonymy are overall significantly larger than other relations across PoS. This indicates that the more frequent antonym slightly tends to precede the less frequent counterpart. For other relations, a preferred order exists at the pair level but is not consistent across pairs, resulting in no clear pattern at the relation level.

Figure 2 shows the distribution of order scores per relation and PoS. For non-antonymy relations, the range of order scores tends to be around 0, confirming that there is no clear co-occurrence order preference[3].

Table 4 shows the co-occurence distance. Antonyms co-occur more closely than other relations. On average, antonyms co-occur within 18 words, which is shorter than the span of pairs in other relations across PoS. Figure 3 shows the distribution of distances per relation and PoS. For antonymy, the third quantile and the maximum of co-occurrence distance are consistently the lowest among all relations across PoS, confirming that antonymous pairs tend to occur close in text.

In short, compared to other relations, antonymy pairs co-occur 1) with greater strength, 2) in the order where the more frequent word tends to precede, and 3) at a shorter distance. Our findings align with previous observations on antonyms, and extend them to other relations and PoS, revealing the distinctive co-occurrence characteristics of antonymy.

---

[3] Note that this finding is based on our definition of relations, in which holonymy and meronymy are not distinguished, nor are hypernymy and hyponymy. If these distinctions are made, holonyms tend to precede meronyms to a moderate degree. Detailed results are provided in Appendix A.

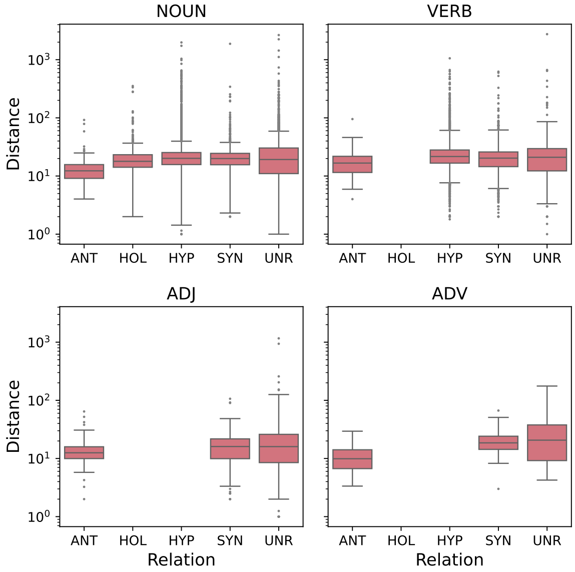


Figure 3: Distribution of distances per PoS and relation with the y axis in logarithmic scale.

5 Discussion

Cross-PoS Analysis The strong co-occurrence of antonymy pairs is often attributed to semantic contrast. Antonymy pairs denote two opposite ends of a scale or a dichotomy in a situation (Cruse, 1986), which might help highlight semantic contrast and hence lead to their significant co-occurrence (Jones, 2006). If the significant co-occurrence is indeed associated with semantic contrast, it should persist after derivation, which usually changes the PoS but not the core meaning. To test this, we examine whether co-occurrence significance is retained after derivation. Among the total of 192 adjectival antonymy pairs, we identify 18 pairs from which 19 non-adjective antonymy pairs are derived (e.g. ("strong","weak") to ("strongly","weakly"))[4]. The 18 original adjectival antonymy pairs co-occur significantly; 18 out of the 19 derived non-adjectival antonymy pairs also exhibit significant co-occurrence. For other PoS and relations, we see a similar pattern. Among all non-adjectival or non-antonymy pairs that significantly co-occur, 85% of their derived pairs sustain the significant co-occurrence. These findings suggest that significant co-occurrence is robust under derivational change, aligning with the idea that co-occurrence might stem from the semantic contrast.

Cross-relation Analysis Antonymy is considered to differ from hypernymy, holonymy, and synonymy in nature because it is not just semantic but also lexically constrained (Justeson and Katz, 1991). For example, "hot" is in antonymy with "cold" but not with "chill". We verify to what extent each relation is lexically constrained. For each relation, we calculate how many lemmas the more frequent lemma is associated with on average [5]. For antonymy, the frequent lemma is associated with only one lemma on average, whereas for other relations, it ranges from 1.7 to 6.7. This result suggests that antonymy involves the strongest lexical pairing, which might enhance their suitability for parallel constructions and, in turn, contribute to their frequent co-occurrence in texts.

Relating Language Models Pretrained language models perform substantially better on antonymy across model architectures and sizes (Pitarch et al., 2023; Cao et al., 2025a). The distinctive co-occurrence characteristics of antonymy may provide signals that facilitate learning antonym pairs during pretraining. However, as prior studies commonly use sentence completion tasks in evaluation, models might take advantage of those intra-sentential co-occurrence characteristics, using them as a shortcut (Du et al., 2023) rather than relying on knowledge of antonymy. Given this, our findings highlight the need to disentangle the extent to which PLMs rely on such distributional clues from the extent to which they generalise beyond them.

6 Conclusion

This work presents an initial exploration of the intra-sentential co-occurrence of antonymy in comparison to other semantic relations across PoS. We find that antonymy is consistently distinctive from all other relations across all metrics used. Our findings establish a robust empirical foundation for the distributional nature of antonymy, offering a solid empirical basis for its future analysis.

Limitations

We only characterise the co-occurrence of semantically related words in a quantitative manner and do not address the qualitative evaluation. We hence can not answer in which lexical and dependency

---

[4] This is done using the information in WordNet. Details can be found in Appendix B.

[5] The Appendix C presents the details.

constructions two words are likely to occur. Furthermore, as an exploratory study, we did not apply word sense disambiguation to a sentence but only used the lemma and PoS of each word as a shallow filtering. We chose not to apply word sense disambiguation in alignment with the previous studies which are done on lemma level. We only studied COCA, and hence, future work should focus on a specific pretraining corpus and examine how PLMs trained on it become biased toward antonymy.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP25KJ1271. We would like to extend our sincere gratitude to Prof. Simone Teufel for valuable discussions. We would also like to express our appreciation to Wanjing Cao, Huaqing Qi, and Shuqin Zhang for their support throughout this work.

References

Edgar Brunner and Ullrich Munzel. 2000. The nonparametric behrens-fisher problem: Asymptotic theory and a small-sample approximation. *Biometrical Journal*, 42(1):17–25.

Zhihan Cao, Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. 2025a. A comprehensive evaluation of semantic relation knowledge of pretrained language models and humans. *Language Resources and Evaluation*.

Zhihan Cao, Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. 2025b. Misalignment of semantic relation knowledge between WordNet and human intuition. In *Global WordNet Conference 2025*.

Walter G. Charles and George A. Miller. 1989. Contexts of antonymous adjectives. *Applied Psycholinguistics*, 10:357–375.

D. Alan Cruse. 1986. *Lexical Semantics*. Cambridge University Press, New York.

Mark Davies. 2008. The corpus of contemporary american english (coca). Data available online at `https://www.english-corpora.org/coca/`(Accessed at 04 May 2025).

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67(1):110–120.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Stefan Evert. 2009. *58. Corpora and collocations*, volume 2, pages 1212–1248. De Gruyter Mouton, Berlin, New York.

Christiane Fellbaum. 1995. Co-occurrence and antonymy. *International Journal of Lexicography*, 8:281–303.

Roxana Girju, Dan Moldovan, and Adriana Badulescu. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32:83–135.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, pages 539–545.

Steven Jones. 2006. A lexico-syntactic analysis of antonym co-occurrence in spoken English. *Text & Talk*, 26(2):191–216.

Steven Jones and M. Lynne Murphy. 2005. Using corpora to investigate antonym acquisition. *International Journal of Corpus Linguistics*, 10(3):401–422.

Steven Jones, Carita Paradis, M. Lynne Murphy, and Caroline Willners. 2007. Googling for 'opposites': a web-based study of antonym canonicity. *Corpora*, 2:129–155.

Steven Jeffrey Jones. 2002. *Antonymy: A Corpus-Based Perspective*. Routledge.

John S. Justeson and Slava M. Katz. 1991. Co-occurrences of antonymous adjectives and their contexts. *Computational Linguistics*, 17:1–20.

Nataša Kostić. 2015. Antonym sequence in written discourse: a corpus-based study. *Language Sciences*, 47:18–31.

Nataša Kostić. 2017. The distributional asymmetries of English antonyms in language use. *Brno studies in English*, 1:[5]–32.

George A. Miller. 1995. Wordnet. *Communications of the ACM*, 38:39–41.

Saif Mohammad, Bonnie Dorr, and Graeme Hirst. 2008. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, Honolulu, Hawaii. Association for Computational Linguistics.

Robert C. Moore. 2004. On log-likelihood-ratios and the significance of rare events. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 333–340, Barcelona, Spain. Association for Computational Linguistics.

Lucia Pitarch, Jordi Bernad, Lacramioara Dranca, Carlos Bobed Lisbona, and Jorge Gracia. 2023. No clues good clues: out of context lexical relation classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5607–5625, Toronto, Canada. Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363. Association for Computational Linguistics.

Sabine Schulte Im Walde and Alissa Melinger. 2008. An in-depth look into the co-occurrence distribution of semantic associates. *Italian Journal of Linguistics*, 20:89–128.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43:781–835.

Shuqiong Wu and Jie Zhang. 2022. Antonym order in english and chinese coordinate structures. *Review of Cognitive Linguistics*, 20(2):530–557.

Appendix

A Ordered Scores for Asymmetric Relations

Figure 4 presents the order scores of asymmetric relations. We redefine the order score for asymmetric relations as follows. A score approaching 1 indicates that a holonym or hypernym precedes its meronym or hyponym, respectively, while a score close to -1 indicates the reverse. For meronymy, more than 50% of the order scores are positive. The results show that holonyms generally precede their meronyms. In contrast, for hypernymy and hyponymy, the order score distribution centers around 0, indicating that the ordering tendency is less clear in the hypernymy/hyponymy relations.

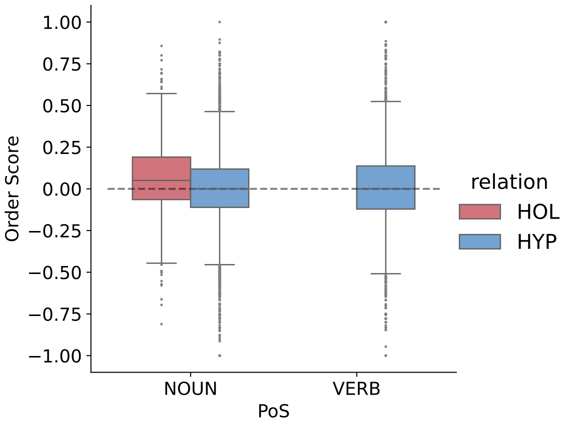


Figure 4: Distribution of order scores of asymmetry relations per PoS.

PoS Orig.	PoS Derv.	Rel. Orig.	Rel. Derv.	Count
ADJ	ADV	ANT	ANT	13 (12)
ADJ	ADV	SYN	SYN	8 (7)
ADJ	NOUN	ANT	ANT	5 (5)
ADJ	NOUN	SYN	SYN	1 (1)
ADJ	NOUN	SYN	HYP	8 (7)
ADJ	VERB	ANT	ANT	1 (1)
ADJ	VERB	SYN	SYN	1 (1)
NOUN	VERB	ANT	ANT	2 (2)
NOUN	VERB	SYN	SYN	15 (12)
NOUN	VERB	HYP	HYP	111 (90)
NOUN	VERB	SYN	HYP	46 (44)
TOTAL				211 (182)

Table 5: Counts of original (Orig.) and derived (Derv.) POS pairs and their corresponding relations. The number of pairs that sustain significance after deriviation are in parenthesis.

POS	ANT	HOL	HYP	SYN	UNR
NOUN	1.0	1.9	5.8	1.5	2.3
VERB	1.1	–	6.9	1.9	1.5
ADJ	1.0	–	–	1.4	1.6
ADV	1.0	–	–	1.5	1.2
Micro AVG	1.0	1.9	6.7	1.7	2.2

Table 6: The average number of lemmas that are associated with the more frequent lemmas per relation and PoS.

B Derived Pairs Statistics

In WordNet, some lemmas are linked to their derived lemmas. For each lemma pair (w, v) that is semantically related and significantly co-occurs, we retrieve all derived lemmas w_d of w and v_d of v using the information in WordNet. For every (w_d, v_d), we check whether it forms a semantic relation and appears in our data. If so, we then examine whether it sustains the co-occurrence significance.

Table 5 presents the results. The semantic usually persists after derivation, particularly for antonymy and hyper-hyponymy. Synonymy pairs sometimes become hyper-hyponymy, reflecting the established similarity between synonymy and hypernymy (Vulić et al., 2017; Cao et al., 2025b).

C Associated Lemma Counts

For each relation, we count how many lemmas are associated with the more frequent lemma on average. In other words, for all pairs (w, v), we retrieve pairs that share w and compute the average number of associated vs per w. Table 6 presents the results.

Antonymy pairs usually have a unique associated antonym for each more frequent lemma w, as the average number of associated lemmas ranges only from 1.0 to 1.1. In holo-meronymy and synonymy, the frequent lemma is associated with slightly more lemmas than in antonymy, with the averages ranging from 1.4 to 2.0. In hyper-hyponymy, the frequent lemma is typically associated with around 6 lemmas. Hence, antonymy is the most lexically constrained among the other relations studied here.

Evaluating Textual and Visual Semantic Neighborhoods of Abstract and Concrete Concepts

Sven Naber[1], Diego Frassinelli[2], and Sabine Schulte im Walde[1]

[1]Institut für Maschinelle Sprachverarbeitung (IMS), Universität Stuttgart
[2]MaiNLP, Center for Information and Language Processing, LMU Munich
{sven.naber, schulte}@ims.uni-stuttgart.de
frassinelli@cis.lmu.de

Abstract

This paper presents a systematic evaluation of nearest neighbors across semantic representation spaces in both textual and visual modalities. We focus on nominal concepts with varying concreteness levels, and apply a neighborhood overlap measure to compare these target concepts differing in their linguistic and perceptual nature. We find that alignment is primarily determined by modality, and additionally by level of concreteness: Models from the same modality show stronger alignment than cross-modal models, and spaces of concrete concepts show stronger alignment than those of abstract ones. Overall, larger neighborhood size strengthens the alignment between spaces.

1 Introduction

Humans possess an intuitive understanding of concepts and their relative positions within a shared semantic space. For instance, we perceive *cat* as being more similar to *dog* than to *table*. This sense of conceptual proximity is grounded in real-world knowledge and similarity across multiple attribute dimensions—such as form (e.g., four-legged animals), category membership (e.g., pets, furniture), and function (Rosch, 1973; Talmy, 1983).

In natural language processing, prototypical attributes and similarity between concepts are generally captured by vector representations and vector distance measures. At the textual level, these representations are learned either through count-based methods such as co-occurrence matrices (Turney and Pantel, 2010), or predictive models such as shallow neural networks trained on target or context prediction objectives (Mikolov et al., 2013; Pennington et al., 2014a). In these vector spaces, relative position and structure encode semantic relatedness. Similarly, vision models such as convolutional neural networks (Krizhevsky et al., 2017) and vision transformers (Dosovitskiy et al., 2020) map images into vector spaces where proximity reflects not only semantic but also perceptual similarity (Battleday et al., 2020). While text embeddings are based on distributional patterns in language, image embeddings are grounded in visual features—such as shape, color, and scene (Krizhevsky et al., 2017).

Level	Range	Concepts
abstract	1.0 – 2.0	*idea*, *justice*
mid-scale	3.0 – 4.0	*story*, *election*
concrete	4.8 – 5.0	*apple*, *car*

Table 1: Concreteness ranges and example concepts based on Brysbaert et al. (2014).

In the current study, we build and compare representations in textual and visual modalities regarding concrete vs. abstract concepts, which differ in their perceptual nature (vision, sound, smell, taste, touch): Concrete concepts such as *apple* are more easily grounded in perceptual features, in contrast to abstract concepts such as *idea*, which lack stable visual referents and are stronger connected to linguistic context (Paivio, 1971; Andrews et al., 2009; Pecher et al., 2011; Frassinelli and Lenci, 2012; Brysbaert et al., 2014; Frassinelli et al., 2017; Naumann et al., 2018; Tater et al., 2024, 2025).

We conduct a systematic analysis of semantic attributes across modalities and across the abstractness-concreteness continuum, by relying on a simple and interpretable nearest-neighbor overlap to capture embedding space alignment between concepts. We demonstrate that modality is indeed the primary factor shaping semantic neighborhoods, with stronger alignment within modalities for both concrete and abstract concepts, while cross-modal agreement is stronger for concrete than for abstract concepts.

2 Data and Methods

In this section, we describe the target concepts, the variants of textual and visual representations, and how we identify nearest neighbors.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 142–151
November 8-9, 2025 ©2025 Association for Computational Linguistics

Target Concepts This study uses a curated set of 1,500 nouns drawn from the Brysbaert concreteness ratings (Brysbaert et al., 2014). To guarantee clear distinctions between sets of concepts, this set consists of 500 nouns each from three distinct levels of concreteness: extremely abstract, intermediate mid-scale, and extremely concrete noun concepts (see Table 1 for concreteness ranges and example concepts). As nearest neighbor candidates for the target concepts we include a larger set of 5,448 nouns from the Brysbaert ratings, as constructed in our previous work (Schulte im Walde and Frassinelli, 2022).

Distributional Word Representations We built textual representations for our target concepts and nearest neighbor candidates by training both count-based and predictive word embedding models on the ENCOW16AX corpus (Schäfer and Bildhauer, 2012; Schäfer, 2015). **Count-based embeddings** were created with a symmetric window of ±20 nouns, verbs and adjectives occurring at least 50 times in the corpus; each word is represented by a 46,716-dimensional vector. **GloVe embeddings** were created using a symmetric window of ±15 and a minimum frequency of 5, relying on the original implementation in Pennington et al. (2014b); each word is represented by a 50-dimensional embedding. **Word2Vec embeddings** were created with a context window of ±5 and a minimum frequency of 5 using the skip-gram objective over 10 epochs with the Gensim package (Řehůřek and Sojka, 2010); each word is represented by a 300-dimensional embedding. **FastText embeddings** were created with a context window of ±5 and a minimum frequency of 2 over 5 epochs using the FAIR FastText implementation (Bojanowski et al., 2017); each word is represented by a 100-dimensional embedding.

Sentence Representations We built a second set of transformer-based textual representations for our target concepts, as a more direct point of comparison with representations based on vision-transformers (see below), by mean-aggregating embeddings of 35 sentences of each noun retrieved from the ENCOW16AX corpus. Embeddings for each sentence were generated from the models with the SentenceTransformer package (Reimers and Gurevych, 2019). **Mpnet embeddings** uses a pre-trained mpnet-base (Song et al., 2020) finetuned on 1B sentence pairs. We use `all-mpnet-base-v2` (Hugging Face, 2021); the resulting embeddings have a dimensionality of 768. **Gemma embeddings** is a 300M parameter embedding model derived from Gemma 3 (Gemma Team et al., 2025). We use `google/embeddinggemma-300m` (Schechter Vera et al., 2025); the resulting embeddings have a dimensionality of 768. **Qwen3 embeddings** use a 0.6B parameter embedding model from the Qwen model family (Yang et al., 2025). We rely on `Qwen/Qwen3-Embedding-0.6B` (Zhang et al., 2025); the resulting embeddings have a dimensionality of 1024.

Visual Representations We built visual representations relying on the top 35 images for each noun using Bing Image Search (Microsoft Corporation, 2025). Images identified as corrupted or irrelevant were automatically replaced. Image quality was probed as described in Appendix A.1. The visual embedding of each concept was created by mean aggregating the `[CLS]` token embeddings, by passing through the respective models the top-n images (with $1 \leq n \leq 35$). **Vision Transformer (ViT)** embeddings use the model `google/vit-base-patch16-224-in21k` (Google, 2021). The resulting embeddings have a dimensionality of 768, serving as a baseline for concept representation from a pure vision transformer trained in a supervised manner. **DINOv2 embeddings** represent concepts using a vision transformer trained in a self-supervised manner. We rely on the `facebook/dinov2-base` model (Oquab et al., 2024; FacebookAIResearch, 2023); the resulting embeddings have a dimensionality of 768. **Hiera embeddings** represent concepts using a vision transformer with a hierarchical architecture trained in a supervised manner. We rely on the `facebook/hiera-large-224-hf` model (FacebookAIResearch, 2024); the resulting embeddings have a dimensionality of 768. **CLIP embeddings** represent concepts using a vision transformer trained to align images and captions with contrastive loss. We rely on the image encoder of the `openai/clip-vit-base-patch32` (OpenAI, 2021; Radford et al., 2021) model; these embeddings have a dimensionality of 512.

Nearest Neighbor Identification We identify the nearest neighbors of a specific concept within a specific semantic embedding space by calculating the cosine between vector representations of the concept and each nearest neighbor candidate in that space, and then sorting the neighbors by decreasing cosine score.

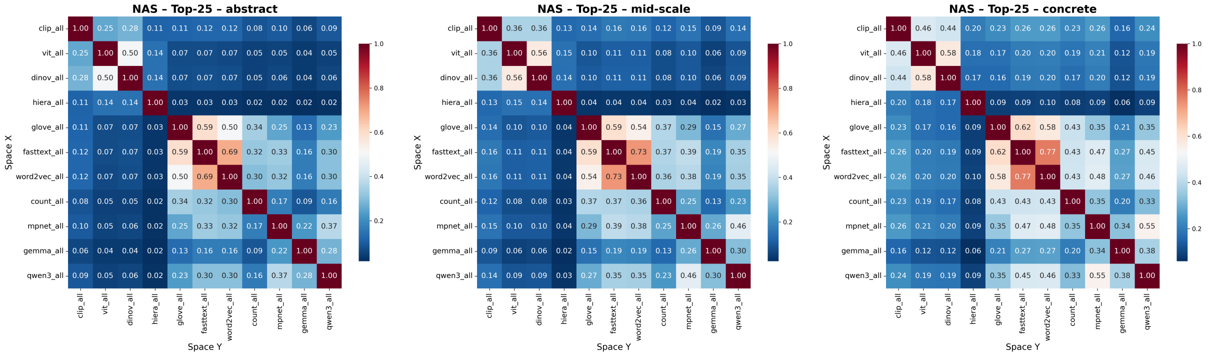


Figure 1: Neighborhood overlap (top-25) across representations for abstract, mid-scale, and concrete concepts.

We vary both the number and start rank of the nearest neighbors we sample, by including the nth to kth nearest neighbors, $1 \leq n \leq k \leq 100$; for example, with $n = 1$ and $k = 5$ we would include the 5 nearest neighbors, and with $n = 5$ and $k = 10$ we would include the 5th to 10th nearest neighbors.

At the core of comparing nearest neighbors across textual and visual semantic representations, we apply a measure of neighborhood overlap in the following way. First, we calculate the overlap $O_c^{n,k}$ of nearest neighbors for a concept c, where $N_{space_i}(c)[n:k]$ denotes the set of neighbors of c ranked from n to k (inclusive) in a given space i:

$$O_c^{n,k} = \frac{|N_{space_1}(c)[n:k] \cap N_{space_2}(c)[n:k]|}{k - n + 1} \tag{1}$$

The concept-specific overlap scores $O_c^{n,k}$ are then aggregated by averaging across all target concepts c in the set C (e.g., the set of all abstract concepts):

$$O_{obs} = \frac{1}{|C|} \sum_{c \in C} O_c^{n,k} \tag{2}$$

In order to interpret these observed overlap scores O_{obs} relative to random chance, we define a normalized alignment score (NAS) by taking into account the expected overlap O_{rand} between two random sets of x neighbors from a candidate set of N concepts. This reflects the expected proportion of shared neighbors for two spaces with random concept positions:

$$O_{rand} = \frac{x}{N - 1} \tag{3}$$

Our NAS score rescales the observed overlap O_{obs} against the expected overlap O_{rand}. A score of 0 indicates overlap by chance; 1 indicates perfect alignment in the observed neighborhood band.

$$\text{NAS} = \frac{O_{obs} - O_{rand}}{1 - O_{rand}} \tag{4}$$

Significance Testing We evaluate whether a single alignment score is significantly greater than chance using a non-parametric sign-flip permutation test which assumes the null hypothesis of symmetric scores around. Given a score vector $\boldsymbol{x} = (x_1, \ldots, x_N)$ across N concepts we compute the observed mean and compare it to a null distribution obtained by randomly flipping signs of x_i across $B = 10{,}000$ permutation. The two-sided p-value is the proportion of permuted means at least as extreme as the observed.

Because our concept sets are large (N=5,448 for the full set of neighbor candidates from Brysbaert, and N=500 for each concreteness band), and because of a standard error of mean scaling with $1/\sqrt{N}$ (Good, 2004), statistically significant alignment differences are at the level of $\overline{\text{NAS}} \approx 0.01$.

3 Experiments and Results

Given that we are interested in overlapping vs. complementary nearest neighbors of abstract and concrete concepts in various modality spaces, we report as results and main insights variants of pairwise comparisons relying on NAS.

Effect of Models and Modalities Figure 1 shows NAS scores for all model pairings across the three concept sets (abstract, mid-scale and concrete, from left to right) at a fixed neighborhood size of 25. The top and left-most four models in the matrices refer to the visual representations, the bottom and right-most seven (four distributional word and three sentence representations) refer to the textual ones.

Modality is clearly the overall dominant factor in alignment scores, i.e., across concreteness levels, model pairs from the same modality (text-text and image-image, cf. top left and bottom right parts of matrices) show substantially stronger overlap than cross-modal

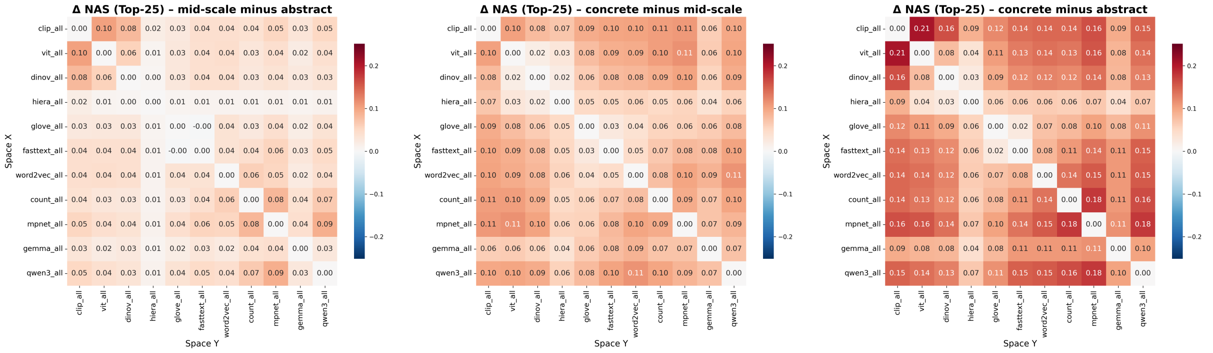


Figure 2: Neighborhood overlap (top-25) differences: $\Delta_{\text{mid}-\text{abstract}}$, $\Delta_{\text{concrete}-\text{mid}}$, and $\Delta_{\text{concrete}-\text{abstract}}$.

pairs: Across abstract/mid-scale/concrete concept sets, alignment scores of image-image spaces reach $0.50/0.56/0.58$ (ViT/DINOv2), and alignment scores of text-text spaces reach $0.69/0.73/0.77$ (Word2Vec/FastText), respectively, while across-modality alignment scores only reach $0.14/0.16/0.26$.

The architectural differences of models within a modality also influence the alignment (with strongest within-modality alignments for word-based distributional models, and weakest ones for sentence-based representations), however to a lesser degree. Overall, the presented alignment patterns thus indicate that while semantic spaces are structured similarly within the same modality, visual and textual modalities shape them in fundamentally different ways.

The modality gap is strongest for abstract concepts, which are inherently more challenging to ground in visual features. For example, the 3×3 top left parts of the matrices in Figure 1 (image-image alignment) show increases from alignment range $[0.25, 0.50]$ for abstract concepts to $[0.36, 0.56]$ for intermediate and $[0.44, 0.58]$ for concrete concepts; the 4×4 middle part of the matrices (distributional text-text alignment, which also presents the overall strongest alignments) shows increases from alignment range $[0.50, 0.69]$ for abstract concepts to $[0.54, 0.73]$ for intermediate and $[0.58, 0.77]$ for concrete concepts.

We also find – but to a more subtle degree – that both within and across modalities the alignment is consistently higher for concrete concepts in comparison to mid-scale concepts, where it is again higher than for abstract concepts. Figure 2 illustrates this pattern across concreteness levels, by showing the differences (Δ scores) between alignment scores of mid-scale vs. abstract concept sets (left), concrete vs. mid-scale sets (middle) and concrete vs. abstract sets (right). We can see that the Δ scores strongly increase from left to right, with differences of up to 0.21 for image-image alignment and up to 0.18 for text-text alignment. This confirms prior research that concrete concepts tend to have less diverse neighbors than abstract concepts (Recchia and Jones, 2012; Kiela et al., 2014; Danguecan and Buchanan, 2016; Reilly and Desai, 2017; Naumann et al., 2018; Schulte im Walde and Frassinelli, 2022; Tater et al., 2024).

Effect of Image Aggregation We now look into the effect of increasing the number of images used for creating visual representations. Figure 3 compares the top-100 overlap of ViT embeddings with an increasing number of images against textual embeddings from FastText, and visual embeddings from DINOv2 and CLIP. The plot illustrates that aggregating image embeddings via mean pooling notably increases alignment scores when adding more images. This development can be attributed to the aggregation process mitigating the multiplicity and saliency issues, which are inherent in any visual representation – challenges which are more pronounced when visual cues are more variable. Accordingly, the aggregation effect is stronger within the vision modality, as it does not affect the gap between the difference in concept representations across modalities. Appendix A.2 further explores the differences of image aggregation across concreteness levels.

Effect of Neighborhood In our last analysis, we explore the role of neighborhood sizes, by computing NAS profiles for increasing neighborhoods. The plots in Figure 4 shows these profiles for selected model pairings within and across modalities as well as across concreteness level: abstract, mid-scale, concrete, and all concepts.

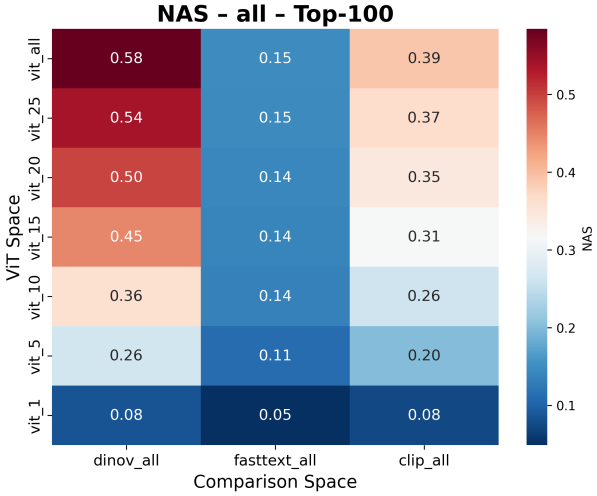


Figure 3: Neighborhood overlap (top-100) across representations and concepts using 1–35 images.

We can see that all alignments involving text models (top three) show a recurring pattern: the alignment is clearly highest for concrete concepts (green lines), and it is lowest for abstract concepts (orange lines), while alignments for mid-scale concepts and across all concepts (red/blue, respectively) are in-between. For the image-image pairing (bottom), the picture is slightly different for concrete concepts, whose alignment declines relative to the other concepts with increasing neighborhood size. Across all pairings, the alignment tends to monotonically increase with larger neighborhood sizes, except in the case of concrete concepts in image-image pairings. The above observations hold for most model pairings.

Overall, our analysis hints at (i) higher variability of immediate neighbors, while larger neighborhoods capture broader semantic similarities, and (ii) more variability in neighborhoods of more abstract in comparison to more concrete concepts. The contrast between within- and cross-modal alignment remains, thus reaffirming the impact of modality on semantic structure.

4 Conclusion

We presented a systematic evaluation of nearest neighbors of abstract and concrete concepts by applying a simple, interpretable, modality- and model-agnostic metric to comparing a variety of textual and visual semantic embedding spaces. Our results confirm that modality is the primary factor shaping semantic structure: Alignments of neighborhoods within the same modality are stronger than alignments across modalities. Specifically focusing on concepts across concreteness levels, we found that concrete concepts show higher alignment of semantic space neighbors than abstract ones, which confirms the difference in perceptual strength in the visual domain, where grounding them is especially challenging. A mean aggregation of images strengthened the alignment with diminishing returns beyond 20–25 images per concept; also, larger neighborhood sizes evoked stronger alignments. Our findings provide a foundation for further analysis of cross-model and cross-modal differences in meaning representation.

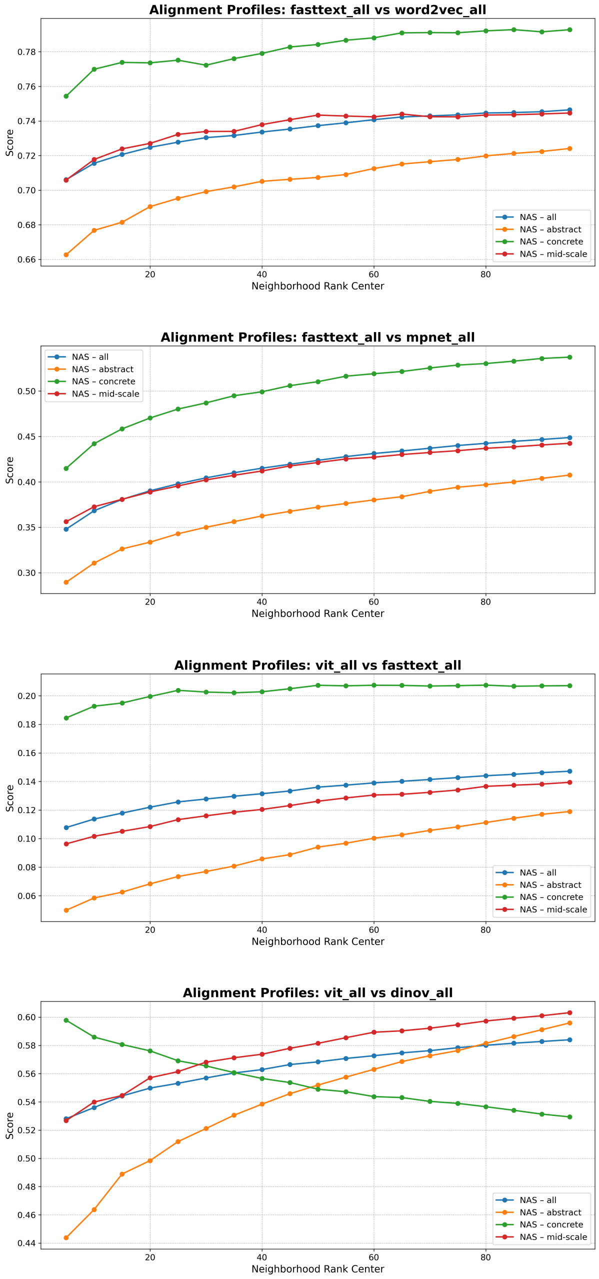


Figure 4: NAS profiles for increasing neighborhood sizes and across concreteness levels.

Limitations

While our method provides interpretable, model- and modality-agnostic comparisons of semantic spaces, several limitations should be acknowledged. Hubness in high-dimensional semantic spaces may inflate alignment scores for some central positioned concepts, which can distort the overlap based comparison for this subset of concepts. Bing Image Search introduces both variation and bias based on cultural and temporal factors (time of retrieval) in its search engine ranking. The retrieved images may therefore not be a comprehensive or completely representative sample of the possible depictions for a target concept. Our approach primarily focuses on local neighborhood structure by comparing ranked orderings. It does not account for global structural properties of these embedding spaces and also overlooks differences in distances between concepts.

Ethical Statement

We do not see any ethical issue related to this work. All our modeling experiments were conducted with open-source libraries, which received due citations.

Use of AI Assistants. The authors acknowledge the use of AI assistants solely for correcting grammatical errors, formatting table boundaries, and providing assistance with coding.

Code and Data Availability

All code and data used in this paper as well as additional plots are openly available at: `https://github.com/SNaber/StarSem2025_SemanticNeighborhoods`

Acknowledgements

This research was supported by the DFG Research Grant SCHU 2580/4-1 *Multimodal Dimensions and Computational Applications of Abstractness*.

References

Mark Andrews, Gabriella Vigliocco, and David Vinson. 2009. Integrating Experiential and Distributional Data to Learn Semantic Representations. *Psychological Review*, 116(3):463–498.

Ruairidh Battleday, Joshua Peterson, and Thomas Griffiths. 2020. Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature Communications*, 11:5418.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness Ratings for 40 Thousand generally known English Word Lemmas. *Behavior Research Methods*, 64:904–911.

Ashley N. Danguecan and Lori Buchanan. 2016. Semantic Neighborhood Effects for Abstract versus Concrete Words. *Frontiers in Psychology*, 7(1034).

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

FacebookAIResearch. 2023. facebook/dinov2-base. `https://huggingface.co/facebook/dinov2-base`. Accessed: 2025-04-28.

FacebookAIResearch. 2024. Hiera: Hierarchical vision transformer models. `https://huggingface.co/facebook/hiera-large-224-hf`. Accessed: 2025-04-28.

Diego Frassinelli and Alessandro Lenci. 2012. Concepts in Context: Evidence from a Feature-Norming Study. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*, Sapporo, Japan.

Diego Frassinelli, Daniela Naumann, Jason Utt, and Sabine Schulte m Walde. 2017. Contextual characteristics of concrete and abstract words. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Phillip I. Good. 2004. *Permutation, Parametric, and Bootstrap Tests of Hypotheses (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Google. 2021. google/vit-base-patch16-224-in21k. `https://huggingface.co/google/vit-base-patch16-224-in21k`. Accessed: 2025-04-28.

Hugging Face. 2021. sentence-transformers/all-mpnet-base-v2. `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`. Accessed: 2025-09-29.

Douwe Kiela, Felix Hill, Anna Korhonen, and Stephen Clark. 2014. Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 835–841, Baltimore, Maryland. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Microsoft Corporation. 2025. Bing image search api. `https://learn.microsoft.com/en-us/bing/search-apis/bing-image-search/overview`.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Daniela Naumann, Diego Frassinelli, and Sabine Schulte im Walde. 2018. Quantitative semantic variation in the contexts of concrete and abstract words. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 76–85, New Orleans, Louisiana. Association for Computational Linguistics.

OpenAI. 2021. openai/clip-vit-base-patch32. `https://huggingface.co/openai/clip-vit-base-patch32`. Accessed: 2025-04-28.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2024. Dinov2: Learning robust visual features without supervision. *Preprint*, arXiv:2304.07193.

Allan Paivio. 1971. Imagery and Language. In Sydney Joelson Segal, editor, *Imagery: Current Cognitive Approaches*, pages 7–32. Academic Press, New York and London.

Diane Pecher, Inge Boot, and Saskia Van Dantzig. 2011. Abstract Concepts. Sensory-Motor Grounding, Metaphors, and Beyond. *Psychology of Learning and Motivation – Advances in Research and Theory*, 54:217–248.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. `https://github.com/stanfordnlp/GloVe`. Accessed: 2025-04-28.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Gabriel Recchia and Michael N. Jones. 2012. The Semantic Richness of Abstract Concepts. *Frontiers in Human Neuroscience*, 6(315).

Megan Reilly and Rutvik H. Desai. 2017. Effects of Semantic Neighborhood Density in Abstract and Concrete Words. *Cognition*, 169:46–53.

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

Eleanor Rosch. 1973. Natural categories. *Cognitive Psychology*, 4(3):328–350.

Roland Schäfer. 2015. Processing and Querying Large Web Corpora with the COW14 Architecture. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*, pages 28–34, Mannheim, Germany.

Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493, Istanbul, Turkey.

Henrique* Schechter Vera, Sahil* Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 69 others. 2025. Embeddinggemma: Powerful and lightweight text representations.

Sabine Schulte im Walde and Diego Frassinelli. 2022. Distributional Measures of Abstraction. *Frontiers in Artificial Intelligence: Language and Computation 4:796756. Alessandro Lenci and Sebastian Padó (topic editors): "Perspectives for Natural Language Processing between AI, Linguistics and Cognitive Science"*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297.

Leonard Talmy. 1983. How Language structures Space. In Herbert L. Pick, Jr. and Linda P. Acredolo, editors, *Spatial Orientation: Theory, Research, and Application*, pages 225–282. Plenum Press, New York and London.

Tarun Tater, Diego Frassinelli, and Sabine Schulte im Walde. 2025. AbsVis – benchmarking how humans and vision-language models “see” abstract concepts in images. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China.

Tarun Tater, Sabine Schulte Im Walde, and Diego Frassinelli. 2024. Unveiling the mystery of visual attributes of concrete and abstract concepts: Variability, nearest neighbors, and challenging categories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21581–21597, Miami, Florida, USA. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. pages 45–50.

A Appendix

A.1 Image Quality Assessment

To assess image quality, we manually rated images for 30 randomly sampled concepts (10 per category) on a Likert-scale based on how well the image visually represented the concept. Images that contained the concept word in written form were counted separately. As shown in Table 2, concrete concepts had the highest visual quality and lowest proportion of textual depictions, while abstract concepts were harder to depict and often appeared as symbolic or textual representations.

Level	Mean Rating	Std Rating	Text
abstract	1.99	0.98	0.39
mid-range	3.59	0.94	0.11
concrete	4.67	0.21	0.01

Table 2: Mean image quality ratings and proportion of images with textual depictions, by concreteness level.

A.2 Effect of Aggregation by Concept Set

In Figure 5 we can see that the effect of aggregation is more pronounced for more abstract concepts, which often do not have direct visual referents and are therefore more variable in their visual representation. In the text-vision comparison the effect is also visible if one accounts for the stronger effect of concreteness levels on final alignment.

A.3 Qualitative Analysis of Nearest Neighbors

We inspected the top-5 nearest neighbors for four concepts of varying concreteness: *eye* (concrete 4.9), *goal* (mid-scale 3.06), *probability* (abstract, symbolically depictable, 1.65) and *ethos* (abstract, symbolically non-depictable, 1.58). Both vision and text models produce plausible neighbors for all concepts, with one exception: lacking clear visual referents, *ethos* poses a challenge for the image models – only CLIP finds meaningful neighbors, as seen in Table 3.

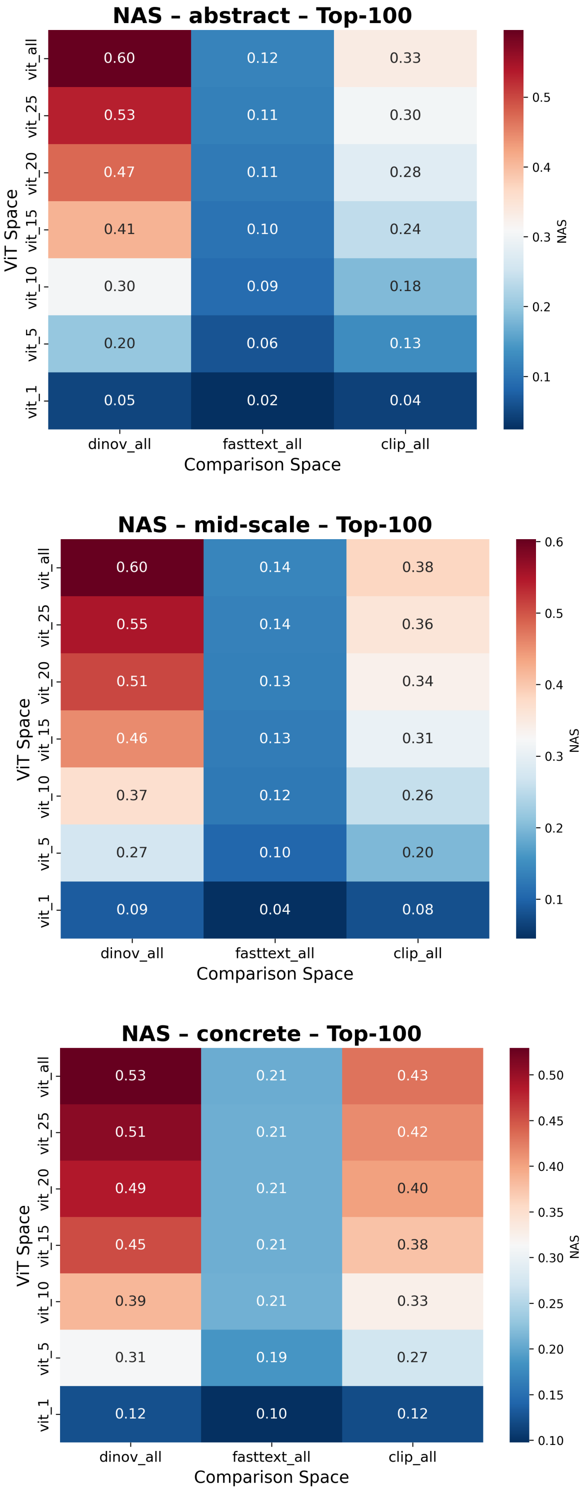


Figure 5: Neighborhood overlap (top-100) across representations for abstract, mid-scale and concrete concepts (top to bottom) using increasingly more images (1–35).

Concept: ***eye***

ViT	**Hiera**	**CLIP**	**Count**	**GloVe**	**FastText**
pupil	cataract	pupil	nose	hair	eyelid
eyelid	pupil	macro	moment	nose	nose
macro	eyelid	eyelid	spectacle	body	eyesight
cataract	macro	blindness	mouth	ear	ear
eyesight	cavity	cataract	amazement	skin	forehead

Concept: ***goal***

ViT	**Hiera**	**CLIP**	**Count**	**GloVe**	**FastText**
competition	spontaneity	effort	ambition	effort	effort
greatness	effort	life	team	chance	scorer
optimism	choice	achievement	effort	success	goalkeeper
determination	inaction	teammate	motivation	advantage	striker
confidence	enthusiasm	victory	ability	momentum	ball

Concept: ***probability***

ViT	**Hiera**	**CLIP**	**Count**	**GloVe**	**FastText**
interpolation	interpolation	interpolation	likelihood	correlation	likelihood
denominator	prevalence	subset	sensitivity	likelihood	variance
differentiation	subroutine	permutation	estimation	variance	estimation
combination	vertex	combination	occurrence	prediction	prediction
fraction	hypothesis	approximation	propensity	estimation	approximation

Concept: ***ethos***

ViT	**Hiera**	**CLIP**	**Count**	**GloVe**	**FastText**
competence	iteration	pathos	academy	ethic	ethic
outcome	proposition	personality	school	mindset	attitude
competency	validity	ethic	community	commitment	commitment
analysis	percentage	trait	pupil	professionalism	tradition
epidemiology	tradeoff	empathy	aspiration	culture	individuality

Table 3: Five nearest neighbors per concept in vision (ViT, Hiera, CLIP) and text (Count, GloVe, FastText) models.

AmbiStory: A Challenging Dataset of Lexically Ambiguous Short Stories

Janosch Gehring
University of Technology Nuremberg
`janosch.gehring@utn.de`

Michael Roth
University of Technology Nuremberg
`michael.roth@utn.de`

Abstract

Word sense disambiguation is the task of selecting a word's applicable word sense in a given context. However, ambiguous texts may lack the information necessary to disambiguate words completely, resulting in multiple word senses with varying degrees of plausibility. We design a dataset around this premise: Our samples consist of 4–5 sentence short stories, where the more plausible word sense of the word to be disambiguated has to be inferred via indirect clues in surrounding sentences. We collect annotations from humans who rate the plausibility of a given word sense on a scale from 1–5. In total, our dataset contains 19,049 human word sense annotations on 1,899 stories. We investigate the performance of large language models on our data and find that many poorly correlate with human judgments. We also find that fine-tuning on our data can increase performance.[1]

1 Introduction

Lexical ambiguity describes the presence of multiple senses being applicable to the same word and has been argued to be a functional property of any efficient communicative system (Piantadosi et al., 2012). Indeed, psycholinguistic studies have shown that humans rapidly process such ambiguities in context (see e.g. McDonald and Shillcock, 2003). Computational approaches to resolving lexical ambiguity in context, commonly referred to as the task of word sense disambiguation (WSD), have been studied at least since the 80s (Dahlgren, 1988; Krovetz and Croft, 1989). In the past decade, research has moved from fixed sense inventories to graded assignments of senses (see §2), following the predominant psycholinguistic view that senses are not (strictly) categorical (for a discussion, see Trott and Bergen, 2023).

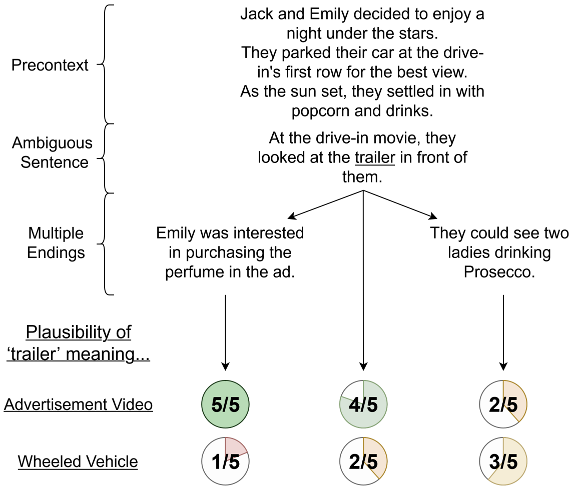

Precontext
Jack and Emily decided to enjoy a night under the stars. They parked their car at the drive-in's first row for the best view. As the sun set, they settled in with popcorn and drinks.
Ambiguous Sentence
At the drive-in movie, they looked at the trailer in front of them.
Multiple Endings
Emily was interested in purchasing the perfume in the ad.
They could see two ladies drinking Prosecco.
Plausibility of 'trailer' meaning...
Advertisement Video
5/5
4/5
2/5
Wheeled Vehicle
1/5
2/5
3/5

Figure 1: An example from our dataset. The ambiguous sentence is re-contextualized in three different endings, resulting in three different stories. Each ending results in different plausibility scores for the word senses.

In parallel with developments in theory formation and corresponding benchmarks, there have been immense technical developments: specifically, large language models (LLMs) such as GPT-4o (OpenAI et al., 2024), DeepSeek (DeepSeek-AI et al., 2025) and Llama-3 (Grattafiori et al., 2024) have not only dominated the NLP community but also shown "superhuman" performance in WSD tasks (Wang and Zhao, 2024). Yet, lexical ambiguities still pose difficulties in tasks such as question answering, natural language inference, and machine translation (Zhang and Choi, 2025). Reasons for this include underspecified language, meaning that lexical ambiguities are not always resolvable (see Haber and Poesio, 2024), as well as differences in background knowledge that can lead to divergent interpretations of context (see e.g. Plank, 2022).

In this paper, we lay the foundations for investigating these difficulties. Specifically, we build a dataset of lexically ambiguous word usages (see Figure 1 for an example) and collect multiple judg-

---

[1] The data will be made available at `https://github.com/Janosch-Gehring/ambistory`.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 152–171
November 8-9, 2025 ©2025 Association for Computational Linguistics

ments on the plausibility of different sentence readings under varying discourse-level contexts. Our dataset consists of 4–5 sentence short stories where the fourth sentence contains a homonym[2] that is used in such a manner that the sentence is ambiguous when read in isolation. Hints towards the preferred reading are provided in surrounding sentences that manipulate the plausibility of word senses by introducing additional details. A correct disambiguation of the homonym thus requires a higher-level understanding of the story. Furthermore, as different contexts will change the plausibility of word senses to varying extents, our dataset contains both stories that are perceived as non-ambiguous and various degrees of ambiguity.

Based on the collected data, we examine the following research questions:

RQ1 How does context affect human plausibility judgments, and when do annotators disagree?

RQ2 How well do judgments by LLMs align with humans, and when are they different or fail?

RQ3 Can an LLM be trained in order to increase agreement with human judgments?

RQ4 How does additional context affect LLM performance and human agreements?

2 Related Work

Word Sense Disambiguation (WSD) has been studied extensively in the NLP community. The 'Word in Context' (*WiC*) series of tasks provide a common framework, in which two occurrences of a *target* word are (typically) classified in a binary manner as either representing the same word sense or different ones (Pilehvar and Camacho-Collados, 2019; Raganato et al., 2020). Similarly, the WiC Target Sense Verification (*TSV*) task presents a word in context with one specific sense, which is to be labeled as correct or incorrect (Breit et al., 2021).

While numerous WSD tasks operate on the sentence-level, a considerable body of research has also been dedicated to document-level WSD. For instance, datasets for 'All-Words' WSD tasks, where the objective is to classify the word sense of every word in the text, commonly consist of longer documents (Moro and Navigli, 2015). Other datasets similarly consist of large tagged corpora (Miller et al., 1993; Taghipour and Ng, 2015).

Comparatively little research has been dedicated to the idea of multiple word senses having varying levels of applicability. Jurgens and Klapaftis (2013) treat WSD as a ranking task, where word senses are given percentages based on their applicability. Other research tackles WSD as a multi-label classification task (Conia and Navigli, 2021) or introduces an ordinal scale to the WiC task (Schlechtweg et al., 2025; Erk et al., 2009). We combine ideas from these tasks and create the first dataset in which the plausibility of different word senses are judged independently given *varying amounts of context* beyond the sentence of a target word.

3 Data Collection

Stories in our dataset consist of 4–5 sentences: three describing the situation ('precontext'), one ambiguous sentence, and optionally an ending. Our collection of stories is a multi-step process in which we collect the precontexts and ambiguous sentences (forming a 'setup'), as well as the endings and the plausibility annotations separately.[3]

We present our data collection approach in the following. First, we extract a set of homonyms to use in our short stories (§3.1). Then, we let humans write ambiguous sentences and prepend a precontext to them to form the setup (§3.2). We then collect ending sentences written by humans (§3.3), and finally collect human plausibility ratings for each 'sample', i.e. the combination of word sense and story (§3.4).

For each step, we collect data via Prolific and require users to pass a qualification test before proceeding to the study to ensure data quality. In addition, we manually check each story for adherence to our guidelines and offensive content, and remove all low-quality submissions. During the collection of plausibility ratings, we include a question from the qualification test as an attention check. We filter annotators who fail this check.

3.1 Homonym Collection

We first collect a selection of homonyms and their word sense pairs around which stories are constructed. The objective is to collect word sense

---

[2] We use the term *homonym* loosely to refer to lexical ambiguities in our data, since most of the relevant sense pairs are presumably unrelated. This does not exclude cases of polysemy, which we also observe to affect annotation. (see §4.2).

[3] All of our data collection was conducted on Prolific under similar conditions, which are further described in Appendix A.1. Screenshots of the annotation interface can be found in Appendix C.

The man crossed the wrong people. He borrowed money from a dangerous group. They came knocking on his door demanding a settlement. The goons made the man **pay**.		Alice loved gardening and enjoyed trying to grow exotic fruit. Jeremy was always excited to taste the fresh produce from their little backyard. Recently, their fruit supply started to dwindle. Alice and Jeremy were running out of fruit, so they went out for a **date**.	
(word sense: *bear a cost or penalty, in recompense of some action*) **Endings:** They stole his belongings and promised to do worse next time before leaving.	(word sense: *give money, usually in exchange for goods and services*) They threatened him until he apologized and returned the money.	(word sense: *a meeting arranged in advance*) **Endings:** They decided to go to a local Hawaiian restaurant that had an extensive dessert menu.	(word sense: *sweet edible fruit of the date palm with a single long woody seed*) Needless to say, there were none to pick, so it necessitated a visit to a large supermarket in the nearby town.

Table 1: Example setups (1st row) from our data, with two different endings for each word sense (2nd row).

pairs where the word senses are distinct, yet can be used within the same sentence structure. Therefore, we decided to extract these from the pun dataset SemEval-2017 Task 7 (Miller et al., 2017), as words used for puns satisfy both criteria. We filter out word sense pairs where the word senses are different parts of speech and ones where a word sense requires a specific particle, as these limit the ambiguous sentences that can be constructed. The rest is used to create a large pool of 729 homonym word sense pairs.

3.2 Writing Ambiguous Sentence

In the second step, we collect ambiguous sentences from crowdworkers. In the annotation interface, humans are tasked with writing sentences where a randomly selected word from our pool of homonyms is used in such a manner that its two displayed word senses are both plausible. We display both word senses and two example sentences, generated with GPT-4o (OpenAI et al., 2024), demonstrating how the word senses can be used in a sentence to help annotators understand technical definitions. If a participant is unable to formulate a sentence, they can click a button to receive a different homonym. Subsequently, we manually filter all sentences that do not conform to the guidelines, e.g. because the homonym is used multiple times or the sentence is clearly non-ambiguous. In a few cases where the sentence is successfully ambiguous but clearly references the wrong word senses, we manually replace the word senses with the correct ones. The guidelines for this task are detailed in Appendix A.2.1. We then further utilize GPT-4o to edit the sentences to rectify spelling errors (see Appendix B.1 for details).

Finally, we employ GPT-4o to generate a precontext, comprising three sentences, for each ambiguous sentence. We instruct it to generate a beginning of the story which does not yet resolve the ambiguous word (for details, see Appendix B.2). The purpose of this additional exposition is to ground the narrative, thereby aiding annotators in writing endings and judging the situation's plausibility.

3.3 Writing Endings

We next collect two endings per setup. We display each story to two annotators and assign them the task of composing an ending that enhances the plausibility of one of the word senses of the homonym. As they are each displayed a different word sense, the plausibilities of word senses will vary between the endings despite the setup being the same. For more details about the task guidelines, please refer to Appendix A.2.2. Examples of ended stories in our dataset are displayed in Table 1.

We purposefully do not filter endings which fail to resolve the ambiguity, which is a common occurrence; recognizing to what extent the endings succeed in the homonym's disambiguation is a part of the challenge. Similarly to the previous step, we use GPT-4o to fix spelling errors and manually filter low-quality submissions. Refer to Appendix B.1 for the prompt.

Furthermore, we include an 'open-ended' story for each setup, which is devoid of an ending sentence and thereby commonly leaves the word sense unresolved. Thus, we obtain three stories for each setup: One for each of the endings, and one without an ending.

3.4 Plausibility Rating

Finally, we collect plausibility ratings for each word sense in the context of a story on a Likert scale ranging from 1 to 5, where 1 signifies that a word sense is inconceivable, while 5 represents unambiguous certainty. Each word sense is annotated by at least five annotators. We also give annotators the option to mark stories as 'nonsensical'.

Section	Avg. length (in words)
Precontext	31.5
Amb. Sentence	9.24
Ending	13.5
Entire Story	49.77

Table 2: Average length of story sections in our dataset.

For this task, only one of the word senses is displayed to the annotators; they have to rely on their own language understanding to perceive which other senses of the homonym may cause a potential ambiguity. We split annotators into 130 groups, each of which annotate 30 word sense–story combinations. Annotators do not receive multiple samples containing variations of the same stories so that each story's annotations remain independent of each other. The guidelines for the task are detailed further in Appendix A.2.3.

3.5 Final Data and Split

For computational experiments (§5), we split the resulting data into a training, development and test set. The sets are split by the homonym used, ensuring that the same target word does not appear across sets. The training set includes 2,280 samples, the development set 588 samples, and the test set 930 samples, for a total of 3,798 samples. Each sample provides the plausibility scores assigned for one word sense of a target word in the given story. For each setup, there are three such stories: one open-ended variant (without explicit ending) and two ended variants with one ending collected for each word sense. Consequently, there are six samples per setup, and 633 setups in total. We collected 19,049 human judgments in sum, with at least five plausibility judgments per sample.

4 Data Analysis

Based on the data collection described in Section 3 we want to analyze under which circumstances people view multiple readings as plausible and when disagreements occur (RQ1). Before that, we discuss statistical properties of the collected data (§4.1) and present a preliminary analysis of the effects of context and word senses on plausibility judgments (§4.2).

4.1 Statistics

Story Length Basic statistics of stories in our dataset are displayed in Table 2. The average story is around 50 words long, although open-ended stories are naturally shorter. The ambiguous sentence itself is typically the shortest part of the story, likely because of the restrictions posed on writers.

Homonyms The selection of homonyms during data collection is influenced by randomness and crowdworker preference. Also, some homonyms with more than two word senses have multiple word sense pairs in our random pool, which increases the odds of drawing them. Therefore, some homonyms appear more often than others. During the data collection process, we removed the most popular homonyms from the random pool to prevent overrepresentation. The most common homonyms with ambiguous sentences in our dataset are *change* (10x), *lousy* (9x), *shot* (7x) and *bars* (7x). In total, our dataset contains 361 different ambiguous word forms (305 different lemma), 411 pairs of word senses, and an average of 1.75 sentences per ambiguous word form.

Inter-Annotator Agreement. We analyze the inter-annotator agreement on our dataset using interval scale Krippendorff's α (Krippendorff, 2004). Our annotators achieve an agreement of $\alpha = 0.506$. We find this to be a reasonable level of agreement, given that our task depends on annotators' own subjective intuition regarding the plausibility of a story and the distinction between word senses.

The average standard deviation per sample is $\sigma = 0.946$. The homonym used seems to greatly influence the human agreement, perhaps due to inherent disagreements about word sense distinctions or the complexity of word sense definitions. For example, the homonym with the highest average standard deviation, *identities* ($\sigma = 1.59$), has abstract and mathematical definitions which may have confused the annotators. As the number of data points per homonym are too low to draw definite conclusions, we will leave further exploration of the effect of specific homonyms to future work.

4.2 Effects of Word Senses and Endings

RQ1: How does context affect human plausibility judgments, and when do they disagree? We investigate to what extent endings affect the perceived plausibility of word senses. To this end, we compare the average scores a word sense receives in different variations of a story. On average, each word sense's score differs by around 0.80 ($\sigma = 0.675$) when an explicit ending is added to the stories. This shows that the additional context has a strong effect on what word sense is perceived as plausible for a target word in a given sentence.

When contrasting the scores word senses receive in one ending versus the other, we find that it varies by around 1.18 ($\sigma = 0.941$) on average. However, this variance differs greatly between stories. Some endings fail to change the perceived plausibility, resulting in mostly unchanged scores, whereas other endings resolve the perceived ambiguity of the story.

Disagreements Human disagreement mostly stems from annotators disagreeing on the extent of a story's ambiguity, thus picking extreme values while others pick middle values. Indeed, about 50% of human ratings are either *1* or *5*. The least picked rating is 3 (15%), which indicates that humans typically have a preference for one of the word senses instead of thinking of multiple as equally plausible. While non-ambiguous stories are typically rated as 1–2 or 4–5 by all humans, ambiguous ones are the cause of much disagreement, with ratings for one word sense sometimes ranging across the entire scale. Although some outliers may be attributed to noise, we believe the following to be two of the most important factors for human agreement:

Word Sense Distinction. As previously stated, humans are not given a word sense inventory when rating plausibilities. Even though our dataset focuses on homonymous word senses, there are occasional instances of related word senses, including literal and figurative usages of words such as *alive* and *drooling*. Perhaps because some pairs of senses are perceived as identical in meaning, most annotators picked a label of *5* on all stories and senses of these words, such as 'be envious' and 'let saliva drivel' for *drooling*.

Word Sense Frequency. Lower-frequency word senses tend to cause disagreement, as humans may disagree on the plausibility of their usage or even forget about them altogether. For instance, in a story containing the sentence '*The blankets in the hotel were pretty lousy*', without additional ending context, all annotators rated the word sense '*very bad*' of *lousy* as a *5*. However, humans are less confident about the lower-frequency word sense '*infested with lice*', rating it as either *2* or *3*. Based on annotator comments, it seems that lower-frequency word senses are often not considered as plausible without supporting information. Approximating sense frequencies based on SemCor (Miller et al., 1993), we indeed find a small but highly significant ($p < 0.01$) correlation between the frequency count of a word sense and the average annotator score, as determined using Spearman's ρ (Spearman, 1904).

5 Computational Experiments

Following the analysis of human annotations in Section 4, we next conduct preliminary experiments on our dataset to answer the remaining research questions outlined in the introduction. We first describe the task setup, models and evaluation metrics used across experiments (§5.1) and then address our research questions regarding the alignment between LMs and human judgments (§5.2) and on the possibility of training LLMs to increase this alignment (§5.3). In context of these experiments, we also analyze the effect of endings as additional context on LLM predictions.

5.1 Experimental Setup

In the first experiment, we investigate the performance of LLMs on our test set without any fine-tuning on the training or development set. Formally, we define the task as follows: Each story text T_i in our data contains a *precontext* c_i, *ambiguous sentence* a_i and an optional *ending* e_i, forming triplets $T_i = \langle c_i, a_i[, e_i]\rangle$. The ambiguous sentence a_i contains a homonymous word form w with two word sense definitions $S_w = \{s_1^w, s_2^w\}$. The task is to predict a plausibility score $p = f(s_j^w \mid T)$ for each sense $s_j^w \in S_w$, where $p \in \{1, 2, 3, 4, 5\}$ and f is a function or model that assigns a score.

We test multiple pre-trained models for this task: **GPT-4o-2024-08-06**, **GPT-4o-mini-2024-07-18** (OpenAI et al., 2024), **o3-2025-04-16** (OpenAI, 2025), **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024), **Mistral-7B-Instruct-v0.3** (Jiang et al., 2023), **Mixtral-8x7B-Instruct-v0.1** (Jiang et al., 2024) and **DeepSeek-V3-0324** (DeepSeek-AI et al., 2025). The GPT and DeepSeek models were accessed through their respective APIs using default settings, whereas the other models were tested using the Huggingface transformers library (Wolf et al., 2020). We set the temperature to 0 for all models.[4] We experiment with zero-shot and few-shot prompting techniques. In the zero-shot setting, we prompt the model with an adjusted version of the annotation guidelines, while for few-shot, we also show the examples displayed to annotators before they start the task.

---

[4] o3 does not allow for setting temperature, so default settings were used.

	Spearman	Acc. w/in SD
Random	0.000	0.454
Majority	N/A	0.558
Llama-3 (0-shot)	0.462	0.663
Mistral (0-shot)	0.382	0.568
Mixtral (0-shot)	0.606	0.634
GPT-4o-mini (0-shot)	0.726	0.726
GPT-4o (0-shot)	**0.756**	0.755
o3 (0-shot)	0.753	0.763
DeepSeek (0-shot)	0.740	**0.790**
Llama-3 (4-shot)	0.491	0.694
Mistral (4-shot)	0.209	0.522
Mixtral (4-shot)	0.607	0.649
GPT-4o-mini (4-shot)	0.737	0.726
GPT-4o (4-shot)	0.742	0.725
o3 (4-shot)	0.742	0.760
DeepSeek (4-shot)	**0.767**	**0.816**
Human Upper Bound	0.834	0.892

Table 3: 'Spearman ρ' and 'Accuracy Within Standard Deviation' scores for different baselines, out-of-the-box LLMs and our human upper bound.

Evaluation metrics One of our main research questions concerns how well LLM predictions align with human judgments (RQ2). As there is no gold standard for this setting, we evaluate models based on their correlation with averaged human judgments as well as a variance-adjusted accuracy measure. Specifically, we calculate the correlation between a model's judgments and the human average using **Spearman's** ρ (Spearman, 1904). As some samples have a clearer consensus than others, we consider their annotators' standard deviation for accuracy. That is, we calculate **Accuracy within Standard Deviation** as the proportion of model predictions that are within standard deviation (at least 1) from the average judgment by annotators.

Baselines and upper bound We use two simple baselines for the task: *Random*, which randomly picks a number between 1 and 5 for every sample, and *Majority*, which picks the majority label (which we found to be 4 on average). We also estimate the best possible performance of humans by evaluating each annotator against the other annotators who received the same sample. We calculate this human upper bound by averaging the scores of the highest-scoring human in each group.

5.2 Experiment 1: Out-of-the-Box LLMs

In the first experiment, we test LLMs out-of-the-box to test alignment with plausibility judgments provided by humans. As indicated by the results in Table 3, there are large differences in performance between models, roughly scaling with model size. However, while most models perform better than the majority or random baseline, the human upper bound is still remarkably higher at a Spearman ρ of 0.834 and an accuracy of 89%. The best performing model is DeepSeek-V3, being the only model to obtain an accuracy of over 80%. The reasoning model o3 performs the second best with an accuracy of 76%. GPT-4o models achieve scores between 72–75%, whereas the Mistral, Mixtral and Llama-3 models all score below 70%.

The effectiveness of zero-shot and few-shot prompting techniques also seems to vary between models. GPT-4o in particular is hurt by the addition of few-shot prompting, whereas models such as Mistral-7B and Llama-8B benefit greatly from the addition of examples.

RQ2: How well do judgments by LLMs align with humans, and when are they different or fail? Given the accuracy gap between models and the human upper bound, it seems there is a sizable difference between human and model judgments. Model judgments are also fairly different between models and prompting techniques. Some examples are displayed in Table 4. We identified the following common error sources on our test set:

'Red Herring' Keywords. In many stories where the setup itself already strongly favors one word sense, endings may be ineffective at swaying the plausibilities perceived by humans despite introducing keywords relating to the less plausible word sense. An example of this is the first story in Table 4, where models such as GPT-4o and DeepSeek-V3 gravitate towards the word *cue* referring to a billiards stick due to the ending mentioning snooker, whereas humans remain unsure about the intended word sense.

Judgment of Open-Ended Stories. As we will discuss further in Section 5.3.2, models seem to differ the most from humans on open-ended stories (i.e. without explicit ending). For example, while all humans and models recognize the ambiguity of the word *shots* in the second example of Table 4, their predictions range from 2–4. Meanwhile, all humans picked a label of *3*, indicating no preference for either word sense.

Story	Word Sense	Human average	Model predictions
He spent years perfecting his craft. Every night, he rehearsed in his small, dimly lit room. Now, it was time to showcase his skills on the big stage. In front of the large audience, table set, he waited for his cue. He was ready to play snooker.	sports implement consisting of a tapering rod used to strike a cue ball in pool or billiards	2.8	[4, 2, 5, 4, 3]
It had been a long week for Emma. She felt overwhelmed by everything happening. On Saturday, she finally decided to do something about it. She took three shots that day.	a small drink of liquor	3.0	[4, 4, 3, 2, 2]
The girl packed her bag for a long adventure. She was excited to explore new places she had never been. Along the way, she faced unexpected challenges that left her feeling different. While on a backpacking trip, the girl ended up in a strange state. She hadn't been there before, but she liked it there and would go again.	the territory occupied by one of the constituent administrative districts of a nation	4.4	[5, 2, 2, 2, 3]

Table 4: Examples of human averages and LLM ratings for word sense plausibility in stories. All pictured model scores were taken from the zero-shot setting (order of scores: GPT 4o, 4o-mini, DeepSeek, Llama-3, Llama-3 + FT).

Word Sense Distinction. LLMs, especially ones trained on huge amounts of online data, are likely influenced by online word sense inventories that were seen during training. Thus, it seems reasonable to assume that LLMs may have knowledge about the sense distinctions of dictionaries such as WordNet (Miller, 1994), whereas most humans do not. Indeed, we find that humans frequently group similar word senses together. In the last example of Table 4, humans unanimously agree on the definition of *state* being correct, since it clearly refers to a territory. However, even then, *state* could realistically refer to both a *province* or a *country*, which are mapped to distinct WordNet senses. Therefore, this is a case where human intuition does not align with a dictionary, which could be a large source for disagreements between models and humans.

5.3 Experiment 2: Fine-Tuning Llama-3 for Plausibility Rating

As a follow-up question, we test whether LLMs can be trained in order to increase agreement with human judgments (RQ3). In the following, we demonstrate the utility of our training and validation set by comparing the performance of the out-of-the-box Llama-3 and Mistral model with versions fine-tuned on our data, which we call '**Llama-3 + FT**'and '**Mistral + FT**'.

5.3.1 Setup

We use LoRA (Hu et al., 2021) to fine-tune the LLama-3.1-8B-Instruct and the Mistral-7B-Instruct-v0.3 model on our training and validation set. During the fine-tuning process, we use a dropout of 0.1, a learning rate of 2e-4, a simulated batch size of 16, rank r of 16, α of 32, the target modules *q_proj* and *v_proj*, and pytorch's AdamW optimizer. We set the model to train for 20 epochs with an early stopping patience of 5. Training stopped after 8 epochs for both models.

When feeding our training data into the model, we first display the story itself, followed by the string: '*In this context, how plausible is it that the meaning of the word {homonym} is {definition} (as in: {example sentence})? Return only the numbered score (1, 2, 3, 4 or 5). Do not return anything else!*'. This mirrors the prompt used during testing, as well as the annotation interface seen by humans. We concatenate the rounded average of human ratings to the input string and only train the model to predict that final number.

5.3.2 Results

Results are displayed in Table 5. We observe a highly significant performance difference between the base models and their fine-tuned counterparts (as determined using Wilcoxon signed-rank test; Wilcoxon, 1945). The performance boost gained from fine-tuning on our data lets the Llama-3 + FT model and the Mistral + FT model achieve Accuracy and Spearman ρ competitive with that of strong models such as GPT-4o or GPT-4o-mini.

RQ3: Can an LLM be trained on our data to improve agreement with humans? Given the large score improvement of the fine-tuned models, we are confident our data can also be useful for improving other models' processing of ambiguities. Note that there is no homonym overlap between the test set and the training/development set, so the

	Spearman	Acc. w/in SD
Random	0.000	0.454
Majority	N/A	0.558
Llama-3 (0-shot)	0.462	0.663
Llama-3+ FT (0-shot)	**0.725**	0.698
Mistral (0-shot)	0.382	0.568
Mistral + FT (0-shot)	0.692	**0.726**
Llama-3 (4-shot)	0.491	0.694
Llama-3 + FT (4-shot)	**0.751**	**0.795**
Mistral (4-shot)	0.209	0.522
Mistral + FT (4-shot)	0.684	0.733
Human Upper Bound	0.834	0.892

Table 5: Results for Llama-3 and Mistral out-of-the-box (LLama-3 / Mistral) and with additional fine-tuning (Llama 3 + FT / Mistral + FT).

All Stories	1	2	3	4	5
Llama-3	0.3	42.5	0.2	56.6	0.5
Llama-3 + FT	25.8	20.4	15.5	32.2	6.1
Human Avg	11.7	21.8	24.2	24.0	18.3
Open-Ended stories only					
Llama-3	0.5	45.2	0.5	53.1	0.8
Llama-3 + FT	26.9	20.3	16.3	32.4	4.0
Human Avg	8.4	23.6	27.1	27.4	13.6
Ended stories only					
Llama-3	0.2	41.2	0.0	58.3	0.3
Llama-3 + FT	25.2	20.4	15.2	32.0	7.2
Human Avg	13.4	21.0	22.7	22.3	20.7

Table 6: Label distribution in the test set (in %). Llama-3 distributions include 0-shot and 4-shot predictions. Human Avg is based on rounded plausibility scores.

	Open-Ended	Ended
Mistral (0-shot)	0.503	0.503
Mistral + FT (0-shot)	0.725	0.726
Mixtral (0-shot)	0.619	0.642
Llama-3 (0-shot)	0.667	0.661
Llama-3 + FT (0-shot)	0.681	0.706
GPT 4o-mini (0-shot)	0.697	0.740
GPT 4o (0-shot)	0.713	0.775
o3 (0-shot)	**0.768**	0.761
DeepSeek (0-shot)	0.755	**0.808**
Mistral (4-shot)	0.635	0.648
Mistral + FT (4-shot)	0.725	0.737
Mixtral (4-shot)	0.626	0.661
Llama-3 (4-shot)	0.739	0.671
Llama-3 + FT (4-shot)	0.774	0.804
GPT 4o-mini (4-shot)	0.694	0.742
GPT 4o (4-shot)	0.7	0.737
o3 (4-shot)	0.781	0.75
DeepSeek (4-shot)	**0.790**	**0.829**
Average (0-shot)	0.681	0.702
Average (4-shot)	0.717	0.731

Table 7: Accuracy within Standard Deviation scores of models on stories without ending (Open-Ended) and with an explicit ending (Ended).

results show that the fine-tuned model generalizes beyond homonyms from the training set.

The improvement is also evident in the change in label distribution, as shown in Table 6 on the example of Llama-3 and Llama-3 + FT. The original Llama-3 model strongly favors the ‘2’ and ‘4’ labels, with those two labels accounting for nearly all of its predictions. In contrast, predictions by Llama-3 + FT are balanced more evenly, aligning much closer with the relatively balanced distribution of the averaged human judgments.

We next perform an additional analysis regarding the role of endings on performance differences between LLMs and human judgments, answering our final research question (RQ4):

RQ4. How does additional context affect LLM performance and human agreements? Intuitively, we expected the disambiguating information found in the endings to remove much of the human disagreement, as there is less need for subjective conjecture. However, while there are some slight tendencies, we do not find the difference between standard deviations of human ratings of open-ended and ended stories to be significant (as determined using Wilcoxon rank-sum test; Wilcoxon, 1945). Nonetheless, we do observe an effect of endings on the label distributions assigned across stories, as displayed in Table 6. In particular, our annotators pick the labels ‘1’ and ‘5’ 5-7 percentage points more often for ended stories than open-ended stories. This effect is much less pronounced for Llama-3 even after fine-tuning, which suggests that there is still room for improvement through other training strategies.

We also investigate whether there is a difference in model performance between ended and open-ended stories. Results are displayed in Table 7.

Interestingly, most models perform better on ended stories than open-ended ones (notable exceptions including o3 and the low-scoring Llama-3). As open-ended samples seem to generally be more challenging to models, but not to humans, we argue that models struggle with the low information content of open-ended stories. Humans may have an innate intuition for interpreting word senses even without evidence that is less pronounced in models. The o3 model does not appear to experience performance decrease on open-ended stories, which suggests that reasoning models may be particularly well-suited for recognizing ambiguity.

6 Conclusion

We introduced *AmbiStory*, a collection of 1,899 short stories with human plausibility ratings for word senses. Each story contains a lexical ambiguity with multiple alternate endings designed to imply different word senses. We investigated how human plausibility judgments are influenced by story context, and found inter-annotator agreement to be affected by the frequency of word senses as well as relations between them, among other reasons. In computational experiments, we found LLM predictions often do not match human-perceived plausibilities, for instance, because of model biases towards high-frequency senses and distracting key words in the story context.

In general, we found models to perform worse when no ending to a story is provided, whereas human agreement remains stable. This may indicate that LLMs lack the common sense to ‘fill in the gaps’ in stories with only minimal disambiguating information. We believe that our dataset provides a useful basis for model development and testing, as exemplified by the possibility of improving the performance of two LLMs via fine-tuning. In fact, AmbiStory also serves as the benchmark for a shared task at SemEval 2026 (Task 5). We encourage the community to use AmbiStory to study and model human perception of lexical ambiguity beyond individual sentences.

Limitations

While we believe AmbiStory to be an effective benchmark in its current state, there are some limitations that we would like to address in future experiments and expansions. Firstly, the stories in this dataset have a fixed five-sentence length and 3/1/1 structure, where the ambiguous sentence is always the fourth sentence and the disambiguating evidence can typically be found in the last sentence. A possible expansion would be to grant crowdworkers more flexibility by adjusting the annotation pipeline, allowing for the story length and the position of the homonym and disambiguating information to be more dynamic (and thus less predictable). Additionally, it would be insightful to collect judgments for domain-specific homonyms and analyze annotator feedback to better understand sources of disagreements.

As an addition limitation, we note that our dataset is currently restricted to English and we collected data primarily from crowdworkers in the UK, whose views may not necessarily reflect those of other native and non-native speakers. Also, as homonym sense pairs are obtained from a preexisting dataset, the set of homonyms in our dataset is not exhaustive. Finally, our stories may contain linguistic biases, such as stylistic tendencies common in language model outputs, as parts of the stories are written or corrected by GPT-4o.

Ethical Considerations

Our dataset contains text written by LLMs and crowdworkers, both of which are susceptible to producing harmful content. We addressed this ethical risk by carefully filtering all samples that could be considered harmful, including for example cases in which one of the word senses has vulgar or offensive connotations in slang. We additionally ensured that crowdworkers are anonymized and paid above minimum wage according to the regulations in the country of the authors.

References

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. WiC-TSV: An evaluation benchmark for target sense verification of words in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1635–1645, Online. Association for Computational Linguistics.

Simone Conia and Roberto Navigli. 2021. Framing Word Sense Disambiguation as a multi-label problem for model-agnostic knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3269–3275, Online. Association for Computational Linguistics.

Kathleen Dahlgren. 1988. Word sense disambiguation.

In *Naive Semantics for Natural Language Understanding*, pages 141–169. Springer.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Katrin Erk, Diana Mccarthy, and Nicholas Gaylord. 2009. Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 10–18.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Janosch Haber and Massimo Poesio. 2024. Polysemy—Evidence from linguistics, behavioral science, and contextualized language models. *Computational Linguistics*, 50(1):351–417.

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

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

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

David Jurgens and Ioannis P. Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *International Workshop on Semantic Evaluation*.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications.

Robert Krovetz and W Bruce Croft. 1989. Word sense disambiguation using machine-readable dictionaries. In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 127–136.

Scott A McDonald and Richard C Shillcock. 2003. Eye movements reveal the on-line computation of lexical probabilities during reading. *Psychological science*, 14(6):648–652.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. o3 and o4-mini system card. `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`. Accessed: 2025-09-11.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. Association for Computational Linguistics.

Dominik Schlechtweg, Tejaswi Choppa, Wei Zhao, and Michael Roth. 2025. CoMeDi shared task: Median judgment classification & mean disagreement ranking with ordinal word-in-context judgments. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 33–47, Abu Dhabi, UAE. International Committee on Computational Linguistics.

C. Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China. Association for Computational Linguistics.

Sean Trott and Benjamin Bergen. 2023. Word meaning is both categorical and continuous. *Psychological Review*, 130(5):1239.

Yuqing Wang and Yun Zhao. 2024. Metacognitive prompting improves understanding in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1914–1926, Mexico City, Mexico. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Michael JQ Zhang and Eunsol Choi. 2025. Clarify when necessary: Resolving ambiguity through interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5526–5543, Albuquerque, New Mexico. Association for Computational Linguistics.

A Annotation Details

A.1 Annotation Setup

We use Prolific for all of our data collection. We set English as first language and an approval rate of at least 97% as prerequisites for participating. Most of our writers and annotators reside in the UK (91%), with some living in the US (5%) and Australia (4%). We provided a median payment of about 11 pounds per hour for all of our tasks, retroactively issuing additional payments where this median was not maintained.

For the 'Ambiguous Sentence Writing Task', annotators are required to write 5 sentences. For the 'Ending Writing Task', they have to produce 20 endings, and for the 'Plausibility Annotation Task', they have to rate 30 samples. We chose the workload per participant to be manageable without them becoming bored or running out of ideas.

A.2 Guidelines

The following are the guidelines, presented on the annotation website in a markdown format.

A.2.1 Ambiguous Sentence Writing Task

Introduction to the Ambiguous Sentence Task

Overview

Your task is to write ambiguous sentences. A sentence is ambiguous if there are multiple ways to understand it.
You will be presented with a word and two of its meanings. For example, the word *bank* and its two word senses "*a financial institution*" and "*slope next to a river*". Your task is to write one sentence where the word is used in such a way that depending on how you choose to interpret it, either meaning could apply.
For example, you could write something like: "*On Saturday morning, I went to the bank.*" The word *bank* here could realistically be either of the two word senses, a financial institution or a slope next to a river, so this would be acceptable.
When writing sentences, please try to uphold the following principles:

- **Ideally, both word senses should be**

equally plausible.

- **Please make sure that the word appears exactly once in the sentence.**
- **The goal is not to write puns, but to write sentences that allow for multiple interpretations.**
- **Submissions that clearly contain AI generated content will be rejected.**

Good Examples

Word: *racket*
Meaning 1: *a loud and disturbing noise*
Meaning 2: *implement used in sports, e.g. tennis racket*
Good Example: *I couldn't concentrate at all because of the loud racket from the tennis game.*
This is good: It could refer to the tennis game itself being loud (meaning 1), or the speaker specifically complains about the sound of the tennis racket (meaning 2). The sentence is plausibly ambiguous.

Word: *season*
Meaning 1: *a set of related television programs*
Meaning 2: *the four times of year (spring, summer, fall, winter)*
Good Example: *Anna thinks that this is the best season.*
This is good: We don't know what "*this*" refers to. It could be either a time of year or a television series. Because of this, the sentence remains successfully ambiguous.

Bad Examples

Word: *bat*
Meaning 1: *an implement with a handle used for hitting the ball in games such as cricket or baseball.*
Meaning 2: *a nocturnal mammal capable of sustained flight*
Bad Example: *The bat flew out of the cave.*
This is bad: It is not plausible that this refers to an implement like a baseball bat. The word "bat" is not used ambiguously.

Word: *root*
Meaning 1: *part of the plant which attaches it to the ground*
Meaning 2: *the basic cause, source or origin of something*
Bad Example: *The root of the problem was buried deep, just like the roots of the old oak tree in the yard.*
This is bad: The word is used multiple times. *Root* should be used only once, and that occurrence should encompass both meanings.

Task Procedure

You will first have to pass a simple 4-question qualification test to confirm that you are human and understand English. The qualification test will ask you to pick the best meaning for a word in a short text. If you fail the qualification test, you will not be able to start writing.
If you pass qualification, the writing page will automatically unlock for you.
If you cannot think of a sentence for the given word, you can press the button on the top of the page to get a new word. You can use the button as many times as you want. You have to write five sentences to clear the task.
Good luck and have fun!

A.2.2 Ending Writing Task

Introduction to the Story Ending Task

Overview

You will see the first four sentences of a five-sentence short story. Your task is to write one sentence that finishes the short story coherently.

Importantly, the fourth sentence contains a word which has multiple meanings. The story is written in such a way that you wouldn't know which meaning is the one intended by the author by the first four sentences.
Your ending sentence has to make the intended meaning more plausible than the unintended meaning. Both meanings will be displayed to you.
Try to not explicitly spell out the meaning. Understanding the intended meaning should only be easy to those who pay attention and understand the story. You can achieve this by *avoiding words closely associated with the intended meaning*, and by *only implying the happenings in the fourth sentence instead of outright stating them.* See below for examples.
The story ending should also be *logical* given the first four sentences. The story does not have to be interesting, it just has to make sense and be coherent.
Also, **your ending sentence should NOT include that ambiguous word with multiple meanings from the fourth sentence.**
Also, feel free to add comments in the comment field.

Examples

Story: “Mr Ellis walked to the town square with a big smile. He carried his easel with him. He was getting ready to paint. Whenever he sets up his easel in the town square, he always draws a crowd.”
Intended Meaning of “draws”: to attract
Unintended Meaning of “draws”: produce an image of something or someone
Your Ending Sentence: Everyone was quite impressed by his picture of a flower.
Explanation: In the first four sentences, it is not clear if he literally sketches a crowd or if he simply attracts a crowd. The ending sentence resolves this by implying that he was sketching a flower, not a crowd.

Story: “The battle raged on as the sun began to set. He crouched behind a crumbling wall, desperately scanning the ground. Bullets whizzed past while his heart pounded loudly in his ears. He realized that he only had one shot’.
Intended Meaning of “shot”: a chance or opportunity to do something
Unintended Meaning of “shot”: a missile from a firearm
Your Ending Sentence: He thought that it was now or never as he attempted to escape from the battle.
Explanation: Some stories like this may make it hard to argue for the intended meaning when the unintended meaning seems more likely to begin with. Try to be creative and come up with a way to increase the intended meaning's plausibility. Here, the implication of him running away and the expression “now or never” implies that the amount of shots in his firearm is less important than his chance/opportunity to get away.

The Annotation Procedure

You will first have to pass the qualification test, which consists of four questions. **You only have one attempt at this.** The qualification test simply asks you to pick the more likely meaning of English words in the context of a sentence. It is mostly to filter out bots and should be no problem to English speakers.
If you fail at the qualification test: You will not be able to start writing. ***IMPORTANT***: **Remember to copy the Screen-Out Completion Code that will be displayed to Prolific.**
If you succeed at the qualification test: Once you pass the qualification test, you will automatically unlock access to the writing page. Each writer is assigned around 20 stories.
Your progress for both qualification and writing is **saved automatically** anytime you press the *NEXT* button.
Good luck and have fun!

Please note that the sentence about explicitly not using the ambiguous word again in the ending was only added after data collection already started, as some annotators misunderstood the task prior to us adding it. We manually filtered endings where the task was trivialized by not adhering to this.

A.2.3 Plausibility Annotation Task

The following are the guidelines presented to annotators during the plausibility annotation step.

Introduction to the Plausibility Annotation Task

Overview

You will see a short text in which one sentence is written in bold. That sentence contains a word that can typically take on multiple different meanings, depending on the context. One of those meanings is shown to you.
Your task is simple: Annotate how plausible a meaning of a word is in the context of the short text using one of five scores:

- **1**: The displayed meaning is not plausible at all given the context.
- **2**: The displayed meaning is theoretically conceivable, but less plausible than other meanings.
- **3**: The displayed meaning represents one of multiple, similarly plausible interpretations.
- **4**: The displayed meaning represents the most plausible interpretation; other meanings may still be conceivable.
- **5**: The displayed meaning is the only plausible meaning given the context.

See below for examples.
You can also mark stories as *nonsensical*, meaning that the text is strange no matter which meaning of the word is used. Even if a story is nonsensical, try to pick whatever makes the most sense to you. You can simply go with your intuition here.
Also, feel free to add comments in the comment field.
There will be times where there is no objectively correct answer. Whatever the case, always look at all of the sentences and carefully think about how plausible each meaning would be.

Examples

“**The bat flew out of the cave.**”
Meaning of “BAT”: A sports implement for hitting balls (e.g. in baseball)
Your Rating: *1*
Explanation: A baseball bat flying out of a cave is inconceivable; it obviously refers to an animal.

“The letter specified where to meet him. **So after reading it, I went to the bank.**”
Meaning of “BANK”: a financial institution
Your Rating: *3*
Explanation: *Bank* could refer to the financial institution, but it could also be a river bank. Neither one seems particularly plausible compared to the other, so your rating should be in the middle.

“The composer often spontaneously had ideas for new melodies. **She writes notes on a sheet of paper.** She can later turn these into a piece.”
Meaning of “NOTES”: a brief written record; a memo
Your Rating *2*
Explanation: *Notes* could conceivably refer to written memos, but given the surrounding sentences, it is more plausible that she is jotting down musical notes.

“Mr Ellis walked to the town square with a big smile. He was getting ready to paint. **Whenever he sets up his easel in the town square, he always draws a crowd.** His painting of a flower looked really realistic!”
Meaning of “DRAWS”: to attract; direct towards itself
Your Rating: *5*
Explanation: Without the last sentence, it is not clear whether the intended meaning of *draws* is *to sketch* or *to attract*. With the additional context that he is sketching a flower

- not a crowd - it becomes clear that *attract* is the only plausible meaning. **Always look at the whole story before making your decision!**

The Annotation Procedure

You will first have to pass the qualification test, which consists of four questions. **You only have one attempt at this.** Carefully look at the sentences and determine the plausibility of meanings.
If you fail at the qualification test: You will not be able to start the annotation. **Remember to copy the Screen-Out Completion Code that will be displayed to Prolific.**
If you succeed at the qualification test: Once you pass the qualification test, you will automatically unlock access to the annotation page. Each annotator is assigned 30 samples.
Your progress for both qualification and annotation is **saved automatically** anytime you press the *NEXT* button.
Good luck and have fun!

A.3 Qualification Test

We include a similar qualification test at each data collection step, where annotators evaluate the plausibility of word senses in four short stories. Note that while the samples are the same, the task differs between the writing tasks and the plausibility rating task: During writing tasks, annotators have to select the more plausible word sense out of two options, whereas during the plausibility rating task, to keep it similar to the actual annotation, only one word sense is displayed and annotators pick on a scale from 1 to 5. The qualification questions are pictured here. The correct answers are underlined; picking a different answer results in immediate disqualification.

Story: The puzzle pieces were scattered across the table. We spent hours on the puzzle, but each piece seemed to fit nowhere. **It was a hard puzzle.**

Most Plausible Meaning of "HARD":
☑ *difficult to understand or solve*
☐ *solid; not soft*

Plausibility of the meaning 'solid; not soft':
1 2 3 4 5

Story: The hotel I stayed at last week was the worst. The room was dirty and there was no one at the reception. **The service there was really lousy.**

Most Plausible Meaning of "LOUSY":
☐ *infested with lice*
☑ *very bad*

Plausibility of the meaning 'very bad':
1 2 3 4 5

Story: Max started sweating in the summer heat. His entire house felt like a sauna. **If he had a fan, the heat would be bearable.**

Most Plausible Meaning of "FAN":
☑ *a device for creating a current of air*
☐ *an enthusiastic follower or admirer*

Plausibility of the meaning 'an enthusiastic follower or admirer':
1 2 3 4 5

Story: Mia and Peter were preparing for the big race. Peter was confident in his talent, so he did not train much. **In the end, Mia beat him in the race.**

Most Plausible Meaning of "BEAT":
☐ *strike violently*
☑ *come out better in a competition, race or conflict*

Plausibility of the meaning 'come out better in a competition, race or conflict':
1 2 3 4 5

B Prompting

We use GPT-4o prompts at several steps of our study, which we describe here.

B.1 Annotation Cleaning

We use GPT-4o twice to improve the formatting, once for the ambiguous sentences and once for the endings. We instruct it to correct formatting errors such as punctuation and capitalization to improve the readibility of the story. We also instruct it to rewrite sentences containing direct speech into third person to facilitate the construction of a story around them. We attempt to restrict GPT-4o from making any other modifications, as any extraneous change could affect the writers' carefully constructed ambiguity. In a few instances, GPT-4o still replaced the ambiguous word with a synonym, which we manually fixed.

The following is the prompt for cleaning the ambiguous sentences:

> Correct the following sentence by fixing the following:
> - Fixing the spelling (e.g. fix typos, capitalize first letter, add punctuation at the end)
> - Fixing the grammar (if necessary)
> - If the sentence is in direct speech (e.g. contains the word "you"), rewrite it to be in third person.
> - DO NOT CHANGE ANYTHING ELSE. Word choices etc must remain authentic to the original.
>
> The sentence is: {sentence}
> Don't return anything other than the corrected sentence.

The following is the prompt for cleaning endings:

> Correct the following sentence by fixing the following:
> - Fixing the spelling (e.g. fix typos, capitalize first letter, add punctuation at the end)
> - Fixing the grammar (if necessary)
> - The sentence follows this sentence: "{sentence}" and should flow coherently.
> - No other stylistic changes are allowed.
>
> The sentence is: {ending}
> Don't return anything other than the corrected sentence. If no changes are necessary, just return the original sentence again.

B.2 Precontext Generation

We use GPT-4o for generating precontexts. The used prompt is as follows:

> Take the following ambiguous sentence, which is the fourth sentence in a 5-sentence short text:
> {sentence}
> In this sentence, the word {homonym} can mean either "{word sense 1}" or "{word sense 2}", depending on the context.
> This sentence is the fourth sentence in a story (out of five total). Write the first three sentences of the story. The first three sentences should serve as an introduction to the story which explains the circumstances of the current situation. Try to use simple sentences. Avoid complicated structures, long sentences and unnecessary information.
> Importantly, the word's meaning is still rather ambiguous even with the context. Both meanings should still be equally as plausible.
> Return these first three sentences, do not return anything else.

B.3 Prediction

We use the following prompt for predicting labels from all LLMs. It is an edited version of the guidelines seen in Appendix A.2.3. Pictured is the few-shot prompt; the zero-shot prompt is identical except for the omission of examples.

> You will see a short text in which one sentence is marked with "**". That sentence contains a word that can typically take on multiple different meanings, depending on the context. One of those meanings is given to you.
> **Your task is simple: Annotate how plausible a meaning of a word is in the context of the short text using one of five scores:**
> * **1**: The displayed meaning is not plausible at all given the context.
> * **2**: The displayed meaning is theoretically conceivable, but less plausible than

other meanings.
* **3**: The displayed meaning represents one of multiple, similarly plausible interpretations.
* **4**: The displayed meaning represents the most plausible interpretation; other meanings may still be conceivable.
* **5**: The displayed meaning is the only plausible meaning given the context.
There will be times where there is no objectively correct answer. Whatever the case, always look at all of the sentences and carefully think about how plausible each meaning would be.
Take a look at the following examples.

——

The bat flew out of the cave. In this context, how plausible is it that the meaning of the word "bat" is "A sports implement for hitting balls (e.g. in baseball)"?
Correct answer: 1

—

The letter specified where to meet him. **So after reading it, I went to the bank.**
In this context, how plausible is it that the meaning of the word "bank" is "a financial institution"?
Correct answer: 3

—

The composer often spontaneously had ideas for new melodies. **She writes notes on a sheet of paper.** She can later turn these into a piece.
In this context, how plausible is it that the meaning of the word "notes" is "a brief written record; a memo"?
Correct answer: 2

—

Mr Ellis walked to the town square with a big smile. He was getting ready to paint. **Whenever he sets up his easel in the town square, he always draws a crowd.** His painting of a flower looked really realistic!"
In this context, how plausible is it that the meaning of the word "draws" is "to attract; direct towards itself"?
Correct answer: 5

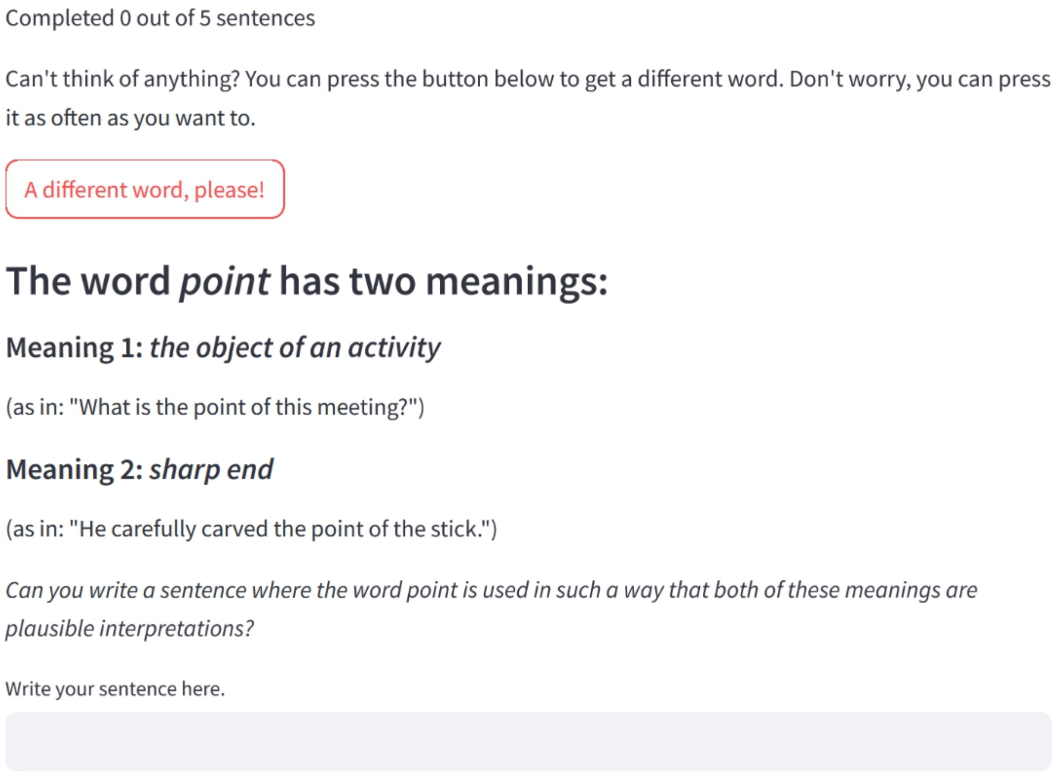
——

Now take a look at the following text: precontext **sentence** ending

In this context, how plausible is it that the meaning of the word "{word}" is "{word sense}"?
Return only the numbered score (1, 2, 3, 4 or 5). Do not return anything else!

C Annotation Interface

Screenshots of our annotation interface can be found in Figure 2, 3 and 4.



Completed 0 out of 5 sentences

Can't think of anything? You can press the button below to get a different word. Don't worry, you can press it as often as you want to.

The word *point* has two meanings:

Meaning 1: *the object of an activity*

(as in: "What is the point of this meeting?")

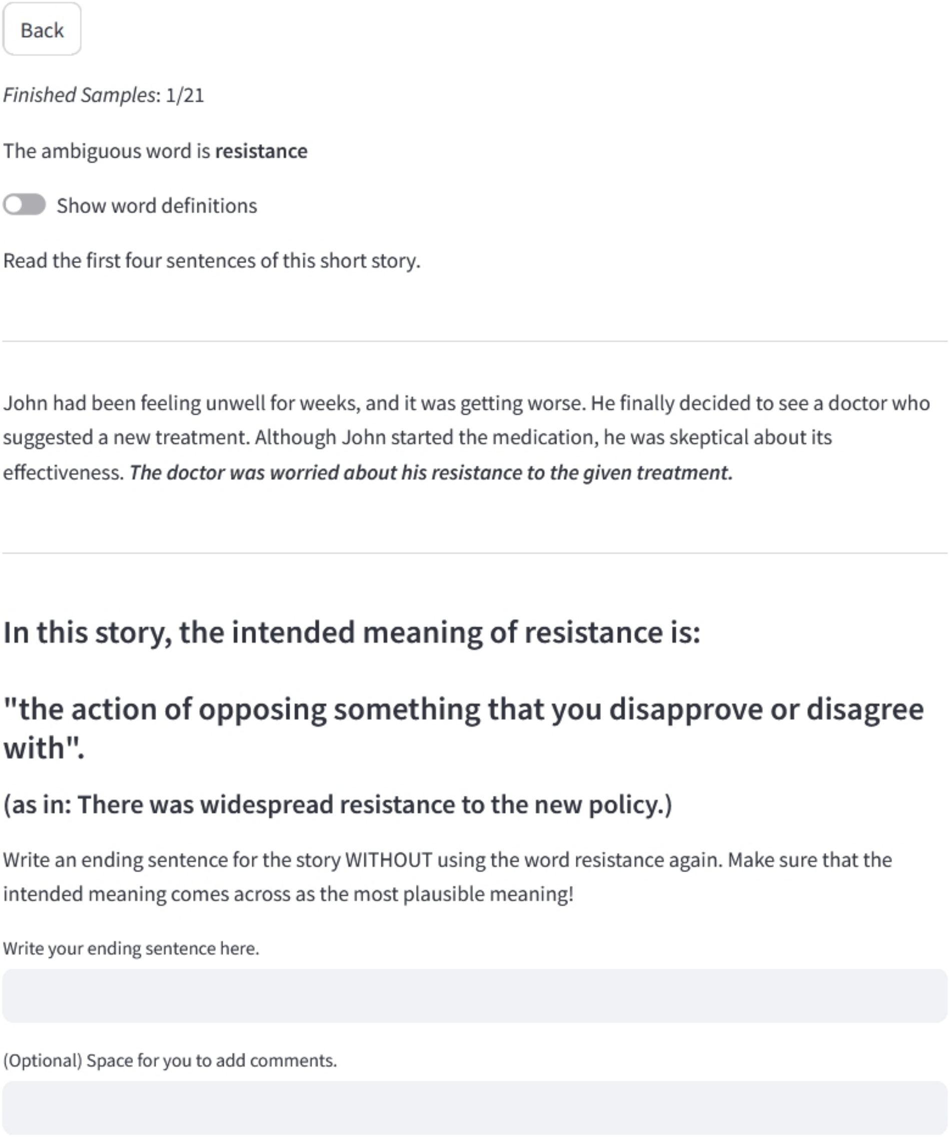
Meaning 2: *sharp end*

(as in: "He carefully carved the point of the stick.")

Can you write a sentence where the word point is used in such a way that both of these meanings are plausible interpretations?

Write your sentence here.

Figure 2: Interface for collecting ambiguous sentences.



Finished Samples: 1/21

The ambiguous word is **resistance**

Read the first four sentences of this short story.

John had been feeling unwell for weeks, and it was getting worse. He finally decided to see a doctor who suggested a new treatment. Although John started the medication, he was skeptical about its effectiveness. ***The doctor was worried about his resistance to the given treatment.***

In this story, the intended meaning of resistance is:

"the action of opposing something that you disapprove or disagree with".

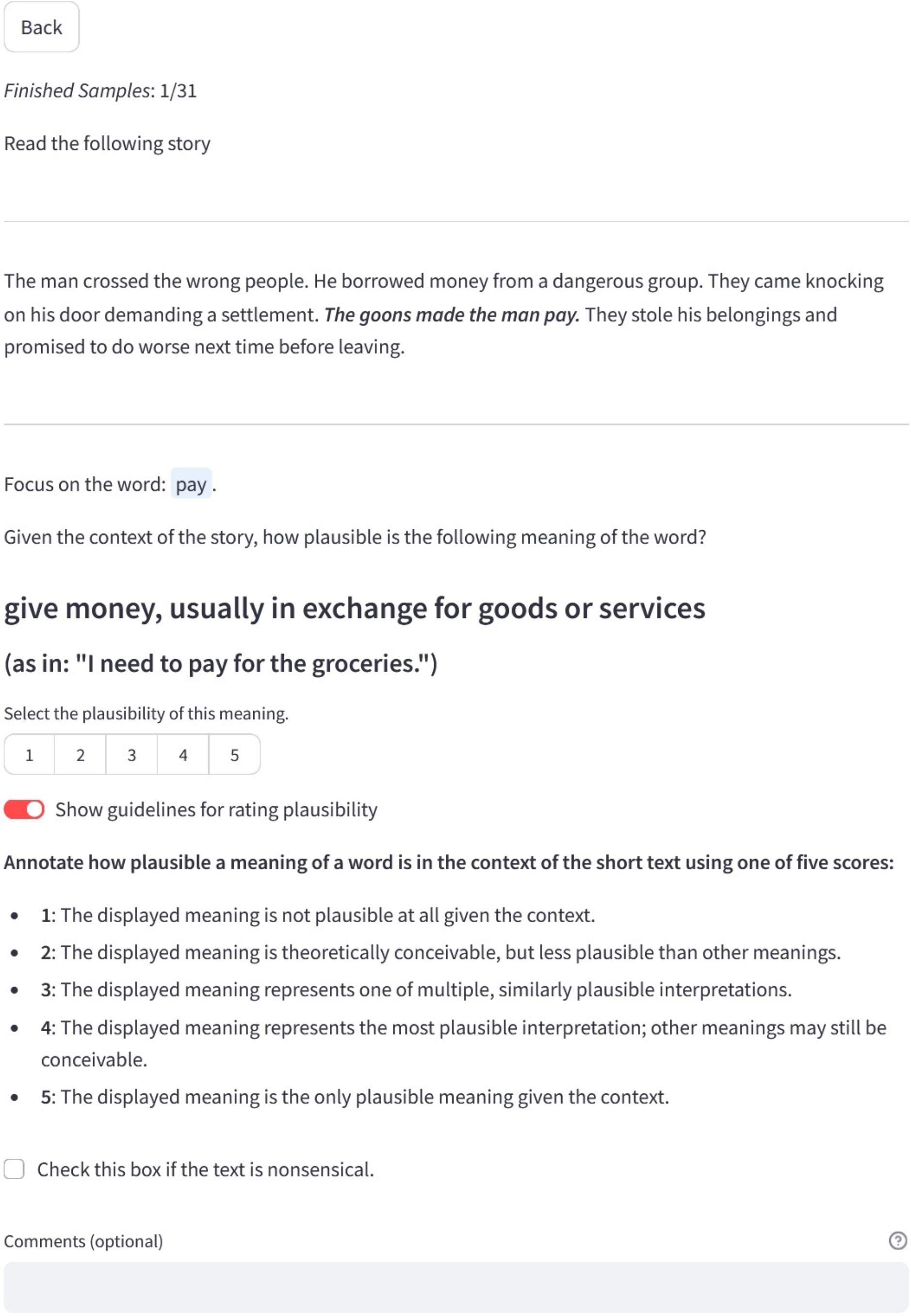
(as in: There was widespread resistance to the new policy.)

Write an ending sentence for the story WITHOUT using the word resistance again. Make sure that the intended meaning comes across as the most plausible meaning!

Write your ending sentence here.

(Optional) Space for you to add comments.

Figure 3: Interface for collecting endings.



Finished Samples: 1/31

Read the following story

The man crossed the wrong people. He borrowed money from a dangerous group. They came knocking on his door demanding a settlement. ***The goons made the man pay.*** They stole his belongings and promised to do worse next time before leaving.

Focus on the word: pay .

Given the context of the story, how plausible is the following meaning of the word?

give money, usually in exchange for goods or services

(as in: "I need to pay for the groceries.")

Select the plausibility of this meaning.

Show guidelines for rating plausibility

Annotate how plausible a meaning of a word is in the context of the short text using one of five scores:

- **1**: The displayed meaning is not plausible at all given the context.
- **2**: The displayed meaning is theoretically conceivable, but less plausible than other meanings.
- **3**: The displayed meaning represents one of multiple, similarly plausible interpretations.
- **4**: The displayed meaning represents the most plausible interpretation; other meanings may still be conceivable.
- **5**: The displayed meaning is the only plausible meaning given the context.

☐ Check this box if the text is nonsensical.

Comments (optional)

Figure 4: Interface for collecting plausibility ratings.

WiC Evaluation in Galician and Spanish: Effects of Dataset Quality and Composition

Marta Vázquez Abuín, Marcos Garcia
CiTIUS – Research Center in Intelligent Technologies
Universidade de Santiago de Compostela
`{martavazquez.abuin,marcos.garcia.gonzalez}@usc.gal`

Abstract

This work explores the impact of dataset quality and composition on Word-in-Context performance for Galician and Spanish. We assess existing datasets, validate their test sets, and create new manually constructed evaluation data. Across five experiments with controlled variations in training and test data, we find that while the validation of test data tends to yield better model performance, evaluations on manually created datasets suggest that contextual embeddings are not sufficient on their own to reliably capture word meaning variation. Regarding training data, our results suggest that performance is influenced not only by size and human validation but also by deeper factors related to the semantic properties of the datasets. All new resources will be freely released.

1 Introduction

Lexical ambiguity (e.g., polysemous words conveying different senses depending on the context) is a central feature of natural languages, and its resolution remains a challenge for computational models, as distinguishing between different senses of a word can be difficult even for humans (Bevilacqua et al., 2021). In NLP, one of the most widely used tasks to evaluate model performance in lexical disambiguation is Word-in-Context (WiC) and its extensions to other languages (Pilehvar and Camacho-Collados, 2019; Raganato et al., 2020), where the goal is to determine whether a target word used in two sentences has the same meaning or not.

These datasets are typically built using existing lexical resources, primarily WordNet and Wiktionaries. As a result, their quality depends heavily on the coverage and reliability of these underlying sources, as well as on the specific methodology used to construct the WiC instances. In this context, human performance on WiC datasets varies considerably, e.g., around 75% in Japanese and 76% in Korean, compared to 97% in Farsi, where sentence pairs were manually grouped by an annotator (Raganato et al., 2020).[1]

In the case of languages such as Galician and Spanish, existing datasets have been automatically constructed using lexical resources with limited coverage, often relying on machine translation both in the development of the WordNets and in the generation of sentence examples (Vázquez Abuín and Garcia, 2025). Moreover, these datasets have neither been validated nor evaluated with respect to human performance, which makes it difficult to assess their overall quality and reliability.

This work investigates the impact of dataset composition and quality on model performance in the WiC task, focusing on both training and evaluation data. We begin by assessing the quality of existing test sets for Galician and Spanish, followed by a validation process to remove instances that are ambiguous for human annotators. Additionally, we construct two new evaluation datasets [2] in which all sentences are manually authored and validated by experts—resources that are also useful for analyzing potential data contamination issues (Sainz et al., 2023). Using these resources, we design five experiments that show that while validating evaluation data tends to improve model performance, results on manually created datasets suggest that contextual embeddings alone are not sufficient to reliably capture word meaning variation. However, during training, data size and human validation seem to have a more limited effect, while other factors related to the semantic properties of the data (such as the well-known effect of word not seen during training) may have a more substantial impact.

---

[1] This variation is also reflected in model performance, where for instance, zero-shot models perform better on Farsi than on Japanese or Korean.

[2] https://github.com/mrtva/wic-eval-starsem25

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 172–178
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Datasets and human annotation

Original datasets: We start by employing, to the best of our knowledge, the only available WiC datasets for Galician and Spanish (Vázquez Abuín and Garcia, 2025). These datasets were constructed following the same methodology as the original WiC paper (Pilehvar and Camacho-Collados, 2019), using WordNets from the Multilingual Central Repository (MCR) (Gonzalez-Agirre et al., 2012).[3] The Galician dataset comprises 1500 training, 400 development, and 1400 test instances. For Spanish, only two splits are available: 200 instances for training or development, and 800 for testing. In all cases, each instance consists of a pair of sentences containing the same word form, along with a binary label indicating whether the word has the same meaning in both contexts ('true') or not ('false'). It is worth mentioning that some instances of the Galician datasets were translated from English or Spanish using machine translation. While the translations were validated by native speakers, the original paper only mentions a manual review of the test data, but does not report any human evaluation of the final datasets.

Validation of the test sets: To assess the quality of the original automatically created dataset, we randomly selected 150 instances from the test sets, which were independently annotated by three experts, bilingual speakers of both languages. Annotators performed the WiC task by determining whether a target word carried the same meaning in two different contexts, assigning a binary label (0 for 'false', 1 for 'true'). No external resources were made available during the annotation process.

The average agreement between each annotator was 71% and 63% for Galician and Spanish, respectively.[4] Table 1 presents the inter-annotator agreement results, both for individual annotator pairs and across all three experts. The agreement levels range from fair to moderate for Galician, and remain fair for Spanish, highlighting potential limitations in automatically generated datasets.

2.1 New human validated test sets

The inter-annotator agreement results indicate that, although automatic dataset creation offers scalability, it may introduce ambiguities or inconsistencies that compromise data quality. To address this, we developed a revised version of the test sets by conducting large-scale human validation and retaining only those instances for which the original and human-assigned labels were in agreement.

Annotators	Gal	Spa
Annotator 1 vs. 2	0.444	0.323
Annotator 2 vs. 3	0.419	0.399
Annotator 1 vs. 3	0.443	0.455
Fleiss' κ	0.435	0.389
Krippendorff' α	0.436	0.390

Table 1: Inter-annotator agreement on the original Galician and Spanish test sets. Top rows are Cohen's κ scores between pairs of annotators, while bottom rows show the $kappa$ and α values for the three annotators.

To this end, we randomly selected 950 instances for Galician and 650 for Spanish from the original test sets, which were then validated by a bilingual language expert. The observed agreement between the expert and the original labels was 71% for Galician and 68% for Spanish. Following the original WiC setup, we ensured a balanced distribution of 'true' and 'false' instances by selecting 450 and 370 sentence pairs for the new validated Galician and Spanish datasets, respectively.

2.2 New manually created test sets

In addition to the potential ambiguities introduced by WordNet examples, automatically constructed datasets also present a risk of data contamination, as many of the sentences may have been seen by language models during pretraining. To assess the impact of these factors, we created an additional test set for each language, composed entirely of manually written sentences. The process is exemplified in Table 2, while Table 3 provides an overview of the three datasets (original, validated, and manually created) for Galician and Spanish.

We randomly selected 50 instances (100 sentences) from the original test sets for each language (*Test set* row in Table 2). Two language experts were then asked to carefully read each sentence, consider the meaning of the target word in its specific context, and compose a new sentence (between 5 and 15 words) in which the target word conveys the same meaning (*Expert* rows). Then, we combined the new sentences produced by each linguist to construct new instances labeled

---

[3] `https://adimen.ehu.eus/web/MCR`

[4] Furthermore, we computed a majority vote accuracy, where each instance was assigned by the majority of the three annotators, with an accuracy of 71.33% for Galician and 62% for Spanish.

Source	**Sentences**		**Label**
Test set	Sentence 1	Sentence 2	False
Expert 1	New_A1_S1	New_A1_S2	—
Expert 2	New_A2_S1	New_A2_S2	—
Comb.	New_A1_S1	New_A2_S1	True
Comb.	New_A1_S2	New_A2_S2	True
Comb.	New_A2_S1	New_A1_S2	False
Comb.	New_A2_S2	New_A1_S1	False

Table 2: Example of the process to create the new dataset. We use sentences from the test set (Sentences 1 and 2) to manually create new contexts with the target word conveying the same meaning (New_A1_S1, etc.).

Dataset	**Train**	**TWs**	**Test**	**TWs**
Original	1500	1187	1400	905
Validated	—	—	450	374
Manual	—	—	172	50
Original	200	190	800	641
Validated	—	—	370	322
Manual	—	—	174	50

Table 3: Summary of the datasets employed in our experiments for Galician (top) and Spanish (bottom), including where each dataset came from (original, validated subset, or manually created), the number of instances per split (train/test) and the number of unique Target Words (TWs). With the exception of the original datasets, both the validated and manually created datasets were reviewed by humans.

as ‘true’—when both sentences were derived from the same original context—and ‘false’—by pairing sentences that originated from different contexts in which the target word had distinct meanings—as can be seen in the *Comb.* rows of Table 2. As before, we maintained a balance between true and false instances by selecting 200 instances per language—100 for each class. Finally, two language experts validated half of the newly created instances, yielding an observed agreement of 86% for Galician and 87% for Spanish. We retained only those instances where there was agreement, resulting in final manually created test sets consisting of 172 sentence pairs for Galician and 174 for Spanish.[5]

3 Experiments

We conducted five experiments to evaluate different combinations of training and test data, including original, human-validated, and fully human-created datasets for Galician and Spanish. These experiments were designed to analyze the impact of composition and quality of the datasets on model performance. The first three experiments use the same original training data while varying the test sets. **Exp1** evaluates performance on the original automatically constructed test set. **Exp2** replaces it with the manually validated version to assess the effect of validation. **Exp3** tests the models on the manually created datasets. Together with the third one, the final two experiments explore the influence of training data on the manually created test sets. In **Exp4**, we augment the original training set with validated data to evaluate the benefit of incorporating human-verified examples. **Exp5** trains models exclusively on the validated datasets to investigate whether smaller, high-quality training sets can outperform larger automatically created ones.

Models: We evaluated base-size encoder models for both Galician—Bertinho (Vilares et al., 2021) and BERT (Garcia, 2021)— and Spanish—Bertin-RoBERTa (De la Rosa et al., 2022) and RoBERTa-BNE (Fandiño et al., 2022).[6] We compared these encoder models to the multilingual XLM-RoBERTa-base (Conneau et al., 2020) in both languages, and included LLaMA 3.2 3B (Grattafiori et al., 2024) as an example of a state-of-the-art multilingual decoder model.

Method: We follow the standard WiC approach proposed by Wang et al. (2019), which involves training a logistic regression classifier on the concatenation of the contextualized representations of the target word in both sentences. For each model, we train a separate classifier for each layer and report the best performance across layers.[7] Word representations are extracted using transformer-based models via the minicons library (Misra, 2022), which is built on HuggingFace’s Transformers.[8]

---

[5] Although other experimental settings with less strict criteria could be explored, in this work we focus on minimizing ambiguities by selecting only agreed cases.

[6] These models were selected because they demonstrate state-of-the-art performance as encoder models for Spanish and Galician across most evaluations.

[7] For each experiment, we evaluated both the original and z-score normalized embeddings, using the standardization method from Timkey and van Schijndel (2021), and report the best results.

[8] `https://github.com/huggingface/transformers`

Baseline: As baselines, we implemented two cosine similarity-based methods: one using the contextualized representations from the transformer models, and another using sentence-level embeddings obtained by averaging FastText embeddings. For both methods, we varied the classification threshold in increments of 0.02, labeling a pair as ‘true’ if the similarity exceeded the threshold.

4 Results and discussion

The best results for each model can be found at Tables 4 and 5, including the highest results for the baselines.[9]

Focusing on the impact of the different test sets, we observe a slight performance improvement when using the human-validated test set in most of the models (Exp1 vs. Exp2), suggesting that human review contributes to the reduction of potential ambiguities and errors of the automatically generated test sets. However, when evaluating on the manually created and validated test sets, we observe a notable performance drop in both languages, particularly for the logistic regression classifiers, and to a lesser extent for the baselines (Exp2 vs. Exp3). This suggests that, despite being less ambiguous, the manually constructed datasets pose greater challenges for models relying on contextualized representations, as simpler cosine similarity-based methods outperform the classifiers.[10] Notably, the baseline results remain largely consistent across different training configurations (Exp3–Exp5 for Galician and Exp4–Exp5 for Spanish), as they share the same test set and the baselines are only minimally influenced by the training data. Contributing factors may include the fact that these sentences were not seen during pre-training, unlike those derived from WordNet and other public resources, which were likely included in the models’ pre-training data, and that they may also differ in nature from examples originating in such lexical resources.

Concerning the impact of incorporating human-validated data into the training (Exp3 vs Exp4), no remarkable changes in the overall performance were observed. However, a deeper analysis reveals that increasing the amount of training data improves generalization, as evidenced by higher accuracy on words not seen during training rises (yielding average gains of 2% for Galician and 3% for Spanish in monolingual models) while only causing minor decreases in performance on seen words.

Finally, although the results from training models exclusively on a small validated dataset (which, in the case of Spanish, is larger than the initial one) are not conclusive, it is noteworthy that in some cases this setup outperforms training on a larger corpus (Exp3), even when including the validated data itself (Exp4). In this respect, and given that the datasets include a range of semantic phenomena (e.g., homonymy, different types of polysemy), these results suggest that generalization may be hindered not just by data size or quality, but also by other factors such as the semantic relatedness between training and test instances or the presence or absence of regular polysemy patterns.

5 Conclusions and further work

This paper presented an evaluation of the impact of dataset composition and quality on WiC performance for Galician and Spanish. We began by assessing the quality of publicly available datasets for these languages, followed by a validation process to enhance the reliability of the test sets. In addition, we constructed new manually created datasets for both languages, also verified by expert annotators. To systematically examine the effects of data quality and composition, we conducted five experiments involving controlled variations in both training and test data.

While models appear to handle many ambiguous cases in the automatically constructed datasets, they often struggle when evaluated on the manually created ones. This suggest that contextualized representations may not fully capture fine-grained sense distinctions, and that simpler methods based on cosine similarity can sometimes be more reliable. Regarding training data, our findings suggest that performance depends not only on the amount and validation of data but also on deeper factors that deserve further analysis, such as the proportion of target words shared between training and test sets, or the semantic relatedness and distribution of polysemy patterns across datasets.

It is worth noting that the models used in this study are general-purpose pretrained models, and

---

[9] The complete results of the baselines are shown in Tables 6 and 7 in Appendix A while the full results by layers and cosine thresholds are reported in Tables 8 and 9 in Appendix B.

[10] Additional evidence for this hypothesis comes from follow-up experiments (not reported here), in which incorporating cosine similarity as a feature into the logistic regression models led to substantial improvements in performance.

Exp	Train	Size	Test	Bas	BERT	Bertinho	XLM	Llama
1	Original	1500	Original	66.4	78.7	78.3	79.6	**81.4**
2	Original	1500	Valid	72.2	79.6	82.2	81.6	**84.0**
3	Original	1500	Manual	**75.7**	53.8	56.7	54.3	52.0
4	Orig+Valid	1950	Manual	**75.7**	56.1	56.1	56.7	55.5
5	Valid	450	Manual	**75.7**	57.2	53.2	56.1	55.5

Table 4: Summary of the best results for each model in Galician across the five experiments. *Bas* is the best baseline (see Table 6 for the complete results).

Exp	Train	Size	Test	Bas	Bertin	RoBERTa	XLM	Llama
1	Original	200	Original	63.1	60.4	61.9	60.5	**64.1**
2	Original	200	Valid	**72.2**	63.2	61.6	61.6	62.7
3	Original	200	Manual	**70.9**	52.0	52.6	53.7	51.4
4	Orig+Valid	570	Manual	**71.4**	54.3	54.3	54.9	55.4
5	Valid	370	Manual	**71.4**	54.9	56.0	56.6	54.2

Table 5: Summary of the best results for each model in Spanish across the five experiments. *Bas* is the best baseline (see Table 7 for the complete results).

not specifically fine-tuned for the WiC task. For future work, we aim to investigate strategies to enhance performance on WiC tasks, ranging from unsupervised methods, such as the WiC-targeted fine-tuning of MirrorWiC (Liu et al., 2021), to supervised fine-tuning approaches exemplified by XL-LEXEME (Cassotti et al., 2023).

Limitations

Models: Regarding the models, our experiments were limited to encoder-based architectures of 'base' and 3B decoder models. As such, the conclusions may not generalize to other types of models, including smaller monolingual decoders or significantly larger multilingual models. Furthermore, all models under consideration are generic pretrained and have not been adapted or fine-tuned specifically for the WiC task.

Data: With respect to the data, some conclusions should be further validated in other languages and with larger datasets. This applies to both training data (which remains limited for Spanish), and evaluation data, especially the manually constructed test sets, which are comparatively small.

Method: As for the evaluation methodology, we rely on standard WiC setups using simple classifiers that operate over concatenated contextual embeddings of the target word. More complex modeling approaches may yield improved results and provide additional insights not captured in this setup.

Analysis: Finally, a more fine-grained analysis would be required to draw robust conclusions about the relationship between training and test corpora. Such an investigation goes beyond the scope of this short paper.

Acknowledgments

This work was funded by MCIU/AEI/10.13039/501100011033 (grants with references PID 2021-128811OA-I00, and CNS2024-154902), by the Galician Government (ED481A-2024-070, ED431F 2021/01, ED431G 2023/04, and ED431B 2025/16), and by a Ramón y Cajal grant (RYC2019-028473-I).

References

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Pierluigi Cassotti, Lucia Siciliani, Marco DeGemmis, Giovanni Semeraro, and Pierpaolo Basile. 2023. XL-LEXEME: WiC pretrained model for cross-lingual LEXical sEMantic changE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1577–1585, Toronto, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Javier De la Rosa, Eduardo G. Ponferrada, Manu Romero, Paulo Villegas, Pablo González de Prado Salas, and María Grandury. 2022. Bertin: Efficient pre-training of a spanish language model using perplexity sampling. *Procesamiento del Lenguaje Natural*, 68(0):13–23.

Asier Gutiérrez Fandiño, Jordi Armengol Estapé, Marc Pàmies, Joan Llop Palao, Joaquin Silveira Ocampo, Casimiro Pio Carrino, Carme Armentano Oller, Carlos Rodriguez Penagos, Aitor Gonzalez Agirre, and Marta Villegas. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68.

Marcos Garcia. 2021. Exploring the representation of word meanings in context: A case study on homonymy and synonymy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3625–3640. Association for Computational Linguistics.

Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. 2012. Multilingual central repository version 3.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2525–2529, Istanbul, Turkey. European Language Resources Association (ELRA).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulic. 2021. Mirrorwic: On eliciting word-in-context representations from pretrained language models. *CoRR*, abs/2109.09237.

Kanishka Misra. 2022. minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models. *arXiv preprint arXiv:2203.13112*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Alessandro Raganato, Tommaso Pasini, José Camacho-Collados, and Mohammad Taher Pilehvar. 2020. Xl-wic: A multilingual benchmark for evaluating semantic contextualization. *CoRR*, abs/2010.06478.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Vilares, Marcos Garcia, and Carlos Gómez-Rodríguez. 2021. Bertinho: Galician BERT Representations. *Procesamiento del Lenguaje Natural*, 66:13–26.

Marta Vázquez Abuín and Marcos Garcia. 2025. Assessing lexical ambiguity resolution in language models with new WiC datasets in Galician and Spanish. *Procesamiento del Lenguaje Natural*, 74:305–319.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Appendix

A Baseline results

Exp	Train	Size	Test	FastT	BERT	Bertinho	XLM	Llama
1	Original	1500	Original	55.6	62.4	**66.4**	60.0	62.1
2	Original	1500	Valid	58.2	69.8	**72.2**	67.6	68.4
3	Original	1500	Manual	57.8	**75.7**	72.8	74.0	68.2
4	Orig+Valid	1950	Manual	57.8	**75.7**	72.8	74.0	67.6
5	Valid	450	Manual	57.8	**75.7**	74.0	74.0	68.8

Table 6: Baseline results for the five experiments (*Exp*) in Galician. *Size* is the number of instances in the training data, while *Train* and *Test* indicate the datasets used.

Exp	Train	Size	Test	FastT	Bertin	RoBERTa	XLM	Llama
1	Original	200	Original	54.1	56.3	60.5	**63.1**	58.6
2	Original	200	Valid	54.3	61.4	69.2	**72.2**	68.9
3	Original	200	Manual	59.4	58.9	**70.9**	64.0	68.6
4	Orig+Valid	570	Manual	59.4	58.9	**71.4**	64.0	68.6
5	Valid	370	Manual	59.4	58.9	**71.4**	64.6	68.6

Table 7: Baseline results for the five experiments (*Exp*) in Spanish. *Size* is the number of instances in the training data, while *Train* and *Test* indicate the datasets used.

B Layers and cosine thresholds for the best results

Exp	Bas		BERT			Bertinho			XLM			Llama		
	Acc.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.
1	66.4	0.60	78.7	9	0.54	78.3	9	0.36	79.6	9*	0.46*	**81.4**	11	0.34
2	72.2	0.58	79.6	9	0.52	82.2	9	0.36	81.6	9	0.52	**84.0**	8	0.34
3	**75.7**	0.32	53.8	10	0.36*	56.7	9	0.30	54.3	11	0.54	52.0	21	0.62
4	**75.7**	0.32	56.1	10	0.54*	56.1	9	0.30	56.7	11	0.52*	55.5	16*	0.52*
5	**75.7**	0.32	57.2	10	0.34*	53.2	7	0.40	56.1	11*	0.52*	55.5	22	0.58

Table 8: Summary of the best results for each model in Galician across the five experiments with their layer(s) and cosine threshold(s). Cells marked with * indicate that multiple layer–cosine combinations yielded the same score; in such cases, we report the configuration with the lowest layer index.

Exp	Bas		Bertin			RoBERTa			XLM			Llama		
	Acc.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.	Acc.	L.	Cos.
1	63.1	0.74	60.4	2	0.74	61.9	10	0.58	60.5	12	0.56*	**64.1**	10	0.30
2	**72.2**	0.52	63.2	3	0.68	61.6	6	0.60	61.6	12	0.62	62.7	8	0.34
3	**70.9**	0.58	52.0	8	0.38*	52.6	9	0.38	53.7	11	0.42	51.4	10	0.32
4	**71.4**	0.58	54.3	8	0.38*	54.3	9	0.38	54.9	11	0.42*	55.4	10	0.32
5	**71.4**	0.58	54.9	8	0.38	56.0	9	0.38	56.6	11	0.44	54.2	10	0.32

Table 9: Summary of the best results for Spanish across the five experiments with their corresponding optimal layer(s) and cosine threshold(s). Cells marked with * indicate that multiple layer–cosine combinations yielded the same score; in such cases, we report the configuration with the lowest layer index.

Math Natural Language Inference: This Should Be Easy!

Valeria de Paiva
Topos Institute
valeria@topos.institute

Qiyue Gao
UC San Diego
q3gao@ucsd.edu

Hai Hu
Shanghai Jiao Tong Univ.
hu.hai@outlook.com

Pavel Kovalev
Carnegie Mellon Univ.
pkovalev@andrew.cmu.edu

Yikang Liu
Shanghai Jiao Tong Univ.
yikangliu@sjtu.edu.cn

Lawrence S. Moss
Indiana Univ.
lmoss@iu.edu

Zhiheng Qian
Shanghai Jiao Tong Univ.
n1vnhil@sjtu.edu.cn

Abstract

We ask whether contemporary LLMs are able to perform natural language inference (NLI) tasks on mathematical texts. We call this the Math NLI problem. We construct a corpus of Math NLI pairs whose premises are from extant mathematical text and whose hypotheses and gold labels were provided by people with experience in both research-level mathematics and also in the NLI field. We also investigate the quality of corpora using the same premises but whose hypotheses are provided by LLMs themselves. We not only investigate the performance but also the inter-group consistency of the diverse group of LLMs. We have both positive and negative findings. Among our positive findings: in some settings, using a majority vote of LLMs is approximately equivalent to using human-labeled data in the Math NLI area. On the negative side: LLMs still struggle with mathematical language. They occasionally fail at even basic inferences. Current models are not as prone to hypothesis-only "inference" in our data the way the previous generation had been. In addition to our findings, we also provide our corpora as data to support future work on Math NLI. Our data can be found at `https://github.com/MathNLI/MathNLI`.

1 Introduction

We study natural language inference (NLI) tasks in the area of research-level mathematics. One might think that LLMs would do extremely well on this task. After all, what counts as an entailment or contradiction in everyday-language texts is often taken as a complicated version of what happens with mathematics. So we might expect purported mathematical inferences to be *easier* to evaluate than those in everyday language. And unlike language in the wild, the domain of mathematics is fairly well-defined. Facts, definitions, and logical reasoning play a large role in mathematical writing. Sentences ought to be precise and unambiguous.

However, there are complications with mathematical text from the start. The vocabulary may be unfamiliar to a generic audience: mathematical parlance can use daily words with new, unfamiliar meanings, e.g. 'ring', 'field', or even 'folklore'. On top of this, the use of visual elements such as symbols, equations, and diagrams, almost changes the very language of the text from plain text to a richer, multimodal language. The field lacks open-source resources such as dictionaries and glossaries for mathematical concepts. It is much harder to find a "person on the street" annotator of mathematics than of more common forms of text.

When confronted with the incredible solutions to mathematical-like problems that deep learning systems can offer nowadays (e.g., AlphaGeometry (Trinh et al., 2024)), it is difficult to believe that these systems cannot understand the basics of causality or of propositional reasoning used throughout mathematics. Nonetheless, when tested on these basics, the LLM-based systems still make very surprising (to humans) mistakes. Further, the fact that LLMs do not have a notion of self-consistency has been documented in many recent papers (Sedova et al., 2024; Kıcıman et al., 2024; Xu et al., 2024). But mathematics, as usually practiced, needs self-consistency. In a sense, it seems that sometimes the deep learning systems deserve an A+ in advanced problem solving but a B in the basics.

For all of these reasons, we could conclude, perhaps surprisingly, that the NLI task is not much easier when using LLMs to deal with mathematical text after all. In this paper, we shall see how precisely correct Math NLI using LLMs can be. We decided to experiment and build a corpus of NLI inference pairs, comparing the output of several LLMs on mathematical text.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 179–188
November 8-9, 2025 ©2025 Association for Computational Linguistics

P (Premise)	H (Hypothesis)	Label
A notion of central importance in categorical topology is that of topological functor.	Topological functor is a notion of categorical topology.	E
The problem of relating a factorization system to a pointed endofunctor is considered.	The problem of relating a factorization system to a pointed endofunctor is not discussed.	C
A notion of central importance in categorical topology is that of topological functor.	There are many notions of central importance in categorical topology.	N

Table 1: Examples in human-created seed Math NLI corpus.

1.1 Research questions

Our big question: Can LLMs be reliable constructors and annotators of Math NLI corpora? We address this by asking and answering some secondary questions: (a) How well do LLMs perform on a Math NLI corpus annotated by mathematicians? (b) Are there common features to the errors which they make? (c) How good is a Math NLI corpus annotated entirely by LLMs? (d) Are LLMs more unanimous on human-written corpora or on corpora generated by LLMs themselves?

1.2 Goal, plan and structure of the paper

The "deliverables" of this paper are two corpora for Math NLI: one written by humans and the other by GPT. These are not benchmarks. But we believe that they will help others who work on this topic.

Equally important, this paper details what we have learned about Math NLI from several years of work, including work that did not turn out as well as we had hoped. Overall, our goal is to make some points about Math NLI which we believe have not been made elsewhere, based on data and examples which we have collected. The plan of the paper is to tell the story of this work.

2 Math NLI seed corpus

2.1 Creation of a seed set of pairs

Our first experiment used a corpus of abstracts of articles in the journal *Theory and Applications of Categories* (TAC) developed in (Collard et al., 2022)[1]. This corpus has some 3K sentences, but 432 were singled out as 'Goldilocks-like sentences': not too short, not too long, and with little or no LaTeX markup. Then we chose 31 of these sentences, and for each sentence S in this set, three of our team members were asked to write a sentence entailed by S, a sentence contradicting S, and a sentence neutral with respect to S. (So we had the "gold labels" by construction. But as we found repeatedly, getting consistent data from humans is difficult, even about mathematical texts.) The team members were told to produce grammatical sentences that did not depend on factual knowledge about the mathematics in the original TAC sentence and that tried to introduce as few new facts as possible. It is impossible to do this perfectly, but the team members strove to do so. We had three people, three labels, and 31 starting sentences. Hence we had $3 \times 3 \times 31 = 279$ pairs, equally divided with E, C, and N labels.

We aimed to fulfill the following conditions as much as possible:

1. Inferences should be uncontroversial. We want inferences which most mathematicians would take to be "immediate."
2. We treat mathematical concepts as black boxes. (Inference should depend as little as possible on the background mathematical knowledge of the assessor.)
3. We avoid "dangling references", pronouns (it, they) or demonstratives (this, that, here, there) without clear antecedents. In general, we tried to avoid all of the problematic issues in natural language semantics.

Table 1 shows some examples of human-created hypotheses and their labels.

Having constructed our seed set of 279 pairs we used a collection of LLMs to evaluate it, as shown in Table 2. This led to the realization that not only did human creators disagree with each other, also the rate of unanimity between machines was not very stable. In particular, we discovered some 20 pairs with contradictory evaluations between machines and humans. We called these the *red pairs*, as they deserved further attention. We explain our process of evaluation, the LLMs used, and our set

---

[1] Available at `https://github.com/ToposInstitute/tac-corpus`.

Abbr.	Model
GPT4	GPT-4[2]
L2	Llama 2 (Touvron et al., 2023) 70B
L3	Llama 3 (Grattafiori et al., 2024) 70B
C3	Claude 3 `https://claude.ai/` unknown
Mistral	Mistral-large
L3.1	Llama-3.1-70B-Instruct (Grattafiori et al., 2024)
Q2	Qwen2-72B-Instruct (Bai et al., 2023)
Mixtral	Mixtral-8x22B-Instruct-v0.1 (Jiang et al., 2024)
DS	deepseek-llm-67b-chat (Bi et al., 2024)
Ge2	gemma-2-27b-it (Team et al., 2024)

Table 2: LLMs used in Exp. 1. Top: Group 1: five initial LLMs; Bottom: Group 2: five later LLMs.

up in the next section, but we discuss briefly the red pairs now.

2.2 Red Pairs

Our three mathematically-trained group members tried to analyze the kinds of mistakes LLMs were making in these pairs. We discovered a few patterns of problematic or flawed reasoning:

Ignored context. Sometimes a specific context was mentioned, for instance

- *P: In the nilpotent case, this nerve is known to be a Kan complex.*
  H: This nerve is not known to be a Kan complex.

but it looks like the LLMs discarded the specific context (*the nilpotent case*) and compared the matrix sentences – in the example above this leads to a contradiction – instead of a neutral label. This is similar to the problems with modal and counterfactual reasoning discussed in (Holliday et al., 2024).

Vague quantifiers. We also have problems with vague predicates like *numerous, few, many*, where humans could also disagree amongst themselves: one example from the 'red pairs' set is

- *P: We worked through numerous examples to demonstrate the power of these notions.*
  H: We worked through two examples to demonstrate the power of these notions.

The mathematicians agreed that *numerous examples* should entail *two examples*, but LLMs did not.

Lexical ambiguity. There is lexical ambiguity, for example, with the verb "resemble" which might mean "is almost equal" (for some humans) or "it looks similar to something else, but it is not the same as" – a reason why we might have humans saying both contradiction or entailment in the example:

- *P: The axioms resemble those for monoidal Abelian categories.*
  H: The axioms are the ones of monoidal Abelian categories.

Note that the ambiguity which we call "lexical" here might also be called "pragmatic" because the issue is whether the use of "resemble" here carries the Gricean implicature that if an object A resembles an object B, then A is not, strictly speaking, B at all.

Naming of math entities. There is a problem with naming mathematical entities, e.g. "group B" vs. "group C" if this is only used as a generic name, as an α-variant, then the difference between B and C doesn't matter. But many times we are talking about different groups.

Unknown math concepts. Sometimes one really must know the concepts involved. For example, for the pair

- *P: This paper proposes a recursive definition of V-n-categories and their morphisms.*
  H: This paper proposes a definition of V-categories.

if we know that 'V-n-categories' are 'V-categories', then we can decide on entailment. But how do we know that? The mathematician is at liberty to create concepts and name them in strange ways. For instance a "skew monoidal category" is not a "monoidal category", only an 'almost' monoidal category.

3 Evaluating LLMs on the seed corpus

In our first experiment, we harness LLMs to evaluate the seed corpus.

3.1 Method

The seed corpus was originally judged by five LLMs, the top ones in Table 2. We used the prompt shown in Appendix C. When 4 or 5 LLMs disagreed with the human annotation, we discussed the pair again, throwing it out if it was considered "controversial" by the mathematicians in our group.

We use API services from `together.ai` to query the LLMs, using a script to extract E/C/N judgments from each model's explanation. The algorithm used is simple: it counts the occurrences of a few keywords in the first sentence without semantical analysis. (It works well if the model gives the answer directly.) However, this algorithm can fail. For example, when the model does not follow the instructions strictly we may end up with a pair that is neither E nor C nor N, and as usual in NLI we take N as a catch-all for “not E and not C.”

3.2 Results

Performance of 10 LLMs on the seed MathNLI corpus is shown in Table 3, with their confusion matrices shown in Table 4.

Table 3 presents the precision, recall, f1-score and accuracy for 10 LLMs. The overall accuracy is medium to high, ranging from 71% to 91%, suggesting that in general, the LLMs we tested can perform category-theory-related mathematical inference to a certain degree. We note that the first group of LLMs (to the left of the table) are not particularly better than the second group (on the right). This might reflect the fact that the first group were closed-source, while Group 2's models were open-source. The first group has two closed source models: Claude 3 and GPT-4; the others are open source. In particular, Claude 3 seems to still be better than the open-source LLMs, but perhaps more runs are necessary to confirm this.

A main message from Table 4 is that most models struggle with *neutral pairs*, mistakenly categorizing them either as entailment pairs or contradictory pairs. For instance, Llama-3 is particularly bad in that it labels as many as 35% of neutral pairs as contradictions; only 48% of the neutral pairs are correctly classified. Claude 3 is the best in labeling N pairs, with an accuracy of 84.9% for them. On the contrary, most models perform very well on C and E pairs. GPT-4, Llama 3 and Qwen2 correctly labeled more than 90% of the C and E pairs. In fact, C pairs are the easiest for all models, except Llama 2, with most models achieving accuracy greater than 90%. Furthermore, models seldom confuse C and E pairs. For eight out of the ten LLMs, C pairs are never categorized as E pairs.

Only one pair in one model (Gemma2) is classified as C by the machines and E by humans:

- *P: Both of them generalize the concept of algebra on a monad T.*
  H: The concept of algebra on a monad T is more special than both of them.

Note that this pair does not satisfy our criteria of explicit references only. The pair is fairly controversial, as well. All LLMs label it as contradictory, but mathematicians tend to think that generalizing and specializing are antonyms. So whatever "both of them" are, if they are a generalization of the concept of algebra of a monad (as claimed by the premise) then "algebra of a monad" will more specialized than them.

Concerning the Group 1 models: out of 279 samples, there is at least one model that agrees with the human annotator in 271 samples. Hence, there are 8 pairs where none of the 5 initial models agrees with the human label. These eight pairs are recalled in Appendix A. The examples are telling as they point out patterns of reasoning that might be difficult for humans as well. For instance:

- *P: Using these ideas, we also prove that magnetic monopoles form an abelian group.*
  H: Using these ideas, we also prove that monopoles form an abelian group.

Clearly a mathematician would gather that ‘magnetic monopoles’ form an abelian group, but nothing has been said about non-magnetic monopoles. So neutral is much more reasonable than ‘entailment’. (More on this is in the appendix A).

Table 5 discusses unanimity between LLMs. As before we consider two groups of models. Our initial LLMs are unanimous in 163 of the pairs (58.4%). Of these 163, in 155 of the cases, the models' agreed-upon label matches the human annotations. And in 271 of the 279 pairs (including ones where the models were not unanimous), at least one model agreed with the human label. This explains the upper row of the table, and the lower row is similar.

Notice that for the more recent LLMs, unanimity goes up from 58.4% to 68.1%. We do not have a good explanation of this.

4 Using LLMs to generate a MathNLI corpus

4.1 Generation using GPT-4

Our second experiment asked GPT-4 to generate Entailment, Contradiction, and Neutral hypotheses from the Goldilocks sentences in the TAC corpus, resulting in 1157 pairs. The prompt we used is shown below:

		GPT4	L2	L3	**C3**	Mistral	L3.1	Q2	Mixtral	DS	Ge2
C	p	82.9	90.5	70.9	91.8	79.8	88.8	87.3	75.4	**92.7**	85.1
	r	**98.9**	61.3	96.8	95.7	97.8	93.5	95.7	**98.9**	81.7	92.5
	f1	90.2	73.1	81.8	**93.7**	87.9	91.1	91.3	85.6	86.9	88.7
E	p	90.1	73.9	85.4	**93.5**	89.8	83.8	80.8	86.3	82.0	82.8
	r	**97.8**	88.2	94.6	92.5	84.9	89.2	90.3	88.2	78.5	82.8
	f1	**93.8**	80.4	89.8	93.0	87.3	86.5	85.3	87.2	80.2	82.8
N	p	**95.5**	56.2	91.8	87.8	81.8	81.7	84.7	85.5	67.6	75.3
	r	68.8	63.4	48.4	**84.9**	67.7	72.0	65.6	57.0	78.5	68.8
	f1	80.0	59.6	63.4	**86.3**	74.1	76.6	73.9	68.4	72.6	71.9
acc		88.5	71.0	79.9	**91.0**	83.5	84.9	83.9	81.4	79.6	81.4
avg	p	89.5	73.5	82.7	**91.0**	83.8	84.8	84.2	82.4	80.8	81.1
	r	88.5	71.0	79.9	**91.0**	83.5	84.9	83.9	81.4	79.6	81.4
	f1	88.0	71.0	78.3	**91.0**	83.1	84.7	83.5	80.4	79.9	81.1

Table 3: Results of 10 LLMs on the seed MathNLI corpus (precision/recall/F1 per class; accuracy and macro averages). *Closed-source* models are marked with the lavender header; green cells denote row-best scores.

> Generate "Entailment", "Contradiction", "Neutral" hypothesis of a given sentence.
> Here are some examples: `[example_script]`
> Sentence: `[context]`

The temperature for the generation was 1. GPT-4 was a good generator of pairs, as we shall see below. But it was not consistent with itself. If it created a pair nominally to be E it could later judge it N or even C. As we see in Table 6, 41.4% of the pairs which GPT-4 created to be neutral it later claims as entailments.

4.2 Checking of a subset, using both humans and LLMs

We chose 89 pairs to conduct manual evaluation and distributed these among the mathematicians of the group. This gave us a set of 89 GPT-4-created/human evaluated pairs. These 89 pairs were also evaluated using GPT-4, Llama 2, Llama 3 and Claude 3, in the first instance. Our mathematicians agree with each other in 80 of the 89 pairs. They agree with 74 (83%) of the GPT-generated labels.

5 Evaluating LLMs on GPT-generated MathNLI corpus

Next, we had the 4 models in Group 1 and 5 models in Group 2 label the 89 pairs. The results are shown in Table 6. The models in group 1 show unanimous agreement in 57 of the pairs (64%), while the models in group 2 do so in 65 (73%). In group 1, for 50 of these 65 pairs (87%), their unanimous label agrees with human labels; while the agreement for group 2 is 57 pairs (88%). Here is our conclusion from this experiment: If we take the unanimous labels from the group 2 models to simply *be* the gold label, then this label is the same as the human label 88% of the time.

The evaluation results on the GPT-generated corpus using the GPT-generated label as the true label are shown in Table 8, with the confusion matrices presented in Table 7. The overall accuracy of LLMs varies between 59.6% and 86.5%, which is relatively lower than the accuracy on the seed corpus.

Our analysis reveals that while the E and C pairs generated by GPT show a certain level of consistency relative to our seed pairs, N pairs are frequently misclassified as E. (This finding echoes what we saw in our previous experiment, but there the pairs were human-generated.) Surprisingly, Llama 2 classifies 75.9% of N pairs as E. Among all evaluated models, Mixtral showed the least susceptibility to this issue, maintaining the highest accuracy of 76.0%. Although its performance on the seed corpus was not outstanding, Mixtral achieved the highest overall accuracy of 86.5% on the GPT-generated corpus.

(a) GPT4				(b) Llama2			(c) Llama3			(d) Claude3			(e) Mistral		
Gold↓	C	E	N	C	E	N	C	E	N	C	E	N	C	E	N
C	98.9	.0	1.1	61.3	1.1	37.6	96.8	.0	3.2	95.7	.0	4.3	97.8	.0	2.2
E	.0	97.8	2.2	.0	88.2	11.8	4.3	94.6	1.1	.0	92.5	7.5	2.2	84.9	12.9
N	20.4	10.8	68.8	6.5	30.1	63.4	35.5	16.1	48.4	8.6	6.5	84.9	22.6	9.7	67.7

(f) Llama3.1				(g) Qwen2			(h) Mixtral			(i) DeepSeek			(j) Gemma2		
Gold↓	C	E	N	C	E	N	C	E	N	C	E	N	C	E	N
C	93.5	.0	6.5	95.7	.0	4.3	98.9	.0	1.1	81.7	.0	18.3	92.5	1.1	6.5
E	1.1	89.2	9.7	2.2	90.3	7.5	3.2	88.2	8.6	2.2	78.5	19.4	1.1	82.8	16.1
N	10.8	17.2	72.0	12.0	21.7	66.3	29.0	14.0	57.0	4.3	17.2	78.5	15.1	16.1	68.8

Table 4: Confusion Matrices Comparison for 10 LLMs on the seed MathNLI corpus. Darker green denotes higher scores, while orange shades denote low scores; for both colors, paler shades represent smaller values in the corresponding range.

	unanimous	some agree w/ human	agrees w/ a human
models in group 1	163 (=58.4%)	271 (=97.1%)	155 (=55.6%)
models in group 2	190 (=68.1%)	266 (=95.3%)	178 (=63.8%)

Table 5: Agreement by LLMs on the seed corpus.

Furthermore, it was observed that LLMs tend to identify C pairs within the GPT-generated corpus more accurately than they do within the seed corpus; 8 out of 9 models achieved an F1 score of over 90%. Notably, Gemma2 successfully detected all the C samples in the GPT-generated corpus.

6 Sanity checks

For previous NLI datasets such as SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018), studies have shown that some of the systems which detect "entailment" were working on artifacts of the datasets, instead of doing any logical reasoning (e.g., Poliak et al., 2018). In this section we provide some sanity checks on our corpus.

6.1 Does every assertion entail itself?

We are trying to make reasoning as clear as possible for humans and systems, and in some sense as simple as possible. We thus propose the easiest logical deduction that we expect LLMs (and humans) to make proficiently. Taking a cue from propositional logic, this would be that every proposition A entails itself, $A \to A$. Informally, if one accepts a sentence S as an assumption, then one should draw S as a trivial conclusion. We want to know whether a computational system "knows" this without being told it explicitly. For an LLM, we want to know if our prompt together with the model's training is enough to carry out what we take to be a very easy inference. To see how much noise LLMs could introduce, we ask them whether every one of a set of mathematical assertions entailed itself. It was not very surprising to discover that for one of 180 such pairs, one LLM missed this assertion $A \to A$.

- *P: We do not include any guiding examples.*
  H: We do not include any guiding examples.

This pair was judged N instead of E. But the explanation for the reasoning of the LLM is fascinating:

> Explanation: In this case, the premise and the hypothesis are identical statements. Since they both convey the same information, there is no logical inference or contradiction between them. The hypothesis is neither entailed by the premise nor does it contradict the premise. Therefore, the hypothesis is neutral to the premise.

The first sentence is correct, but immediately following that the model blunders.

The experiment was repeated with many more pairs, and only 3 in 2312 pairs had the same problem. This is a very small number of errors, but this is arguably the easiest inference one can ever make. It is odd that a system which can make deep, complicated mathematical inferences, can get a simple inference like 'A implies A' wrong.

	unanimous	agree w/ at least 1 human	agree w/ all human
human annotator	/	/	80 (= 89.9%)
GPT generator	/	74 (= 83.1%)	65 (= 73.0%)
models in group 1	57 (= 64.0%)	50 (= 56.2%)	43 (= 48.3 %)
models in group 2	65 (= 73.0%)	57 (= 64.0%)	50 (= 56.2%)

Table 6: Experiment 3 Result: total 89 pairs generated by GPT-4

(a) GPT-4

Gold↓	C	E	N
C	96.7	.0	3.3
E	.0	96.7	3.3
N	.0	41.4	58.6

(b) Llama 2

C	E	N
53.3	.0	46.7
.0	100.0	.0
.0	75.9	24.1

(c) Llama 3

C	E	N
96.7	.0	3.3
.0	100.0	.0
.0	51.7	48.3

(d) Claude 3

C	E	N
93.3	3.3	3.3
.0	93.3	6.7
3.4	34.5	62.1

(e) Llama 3.1

C	E	N
93.3	.0	6.7
.0	100.0	.0
3.4	55.2	41.4

(f) Qwen2

C	E	N
93.3	.0	6.7
.0	100.0	.0
3.4	34.5	62.1

(g) Mixtral

C	E	N
96.7	.0	3.3
.0	96.7	3.3
3.4	31.0	65.5

(h) Deepseek

C	E	N
83.3	.0	16.7
.0	90.0	10.0
.0	31.0	69.0

(i) Gemma2

C	E	N
100.0	.0	.0
.0	90.0	10.0
.0	34.5	65.5

Table 7: Confusion Matrices on GPT-generated Corpus

Previous work such as Xu et al. (2024) tries to catalog the kinds of mistakes that LLMs are known to make. They suggest that “to uncover the logical flaws of LLMs, problematic cases will be attributed to five error types from two dimensions, i.e., *evidence selection process* and *reasoning process*." The example above seems clearly a reasoning process kind of error, as the LLM is very clear that both the hypothesis and the premise are ‘identical statements’. But from that it concludes that the hypothesis is **not** entailed by the premise.

6.2 Contradictions must be symmetric

Most humans would agree that if a sentence A is contradictory with a sentence B, then sentence B is contradictory with A. That is, being contradictory is a symmetric property. Work in (Kalouli et al., 2017) showed that the humans annotating the corpus SICK did not realize when they had non-symmetric contradictions. We hence checked whether LLMs evaluated contradictions symmetrically in the GPT-generated corpus. This small experiment showed that out of 495 pairs (5 times 93 contradiction pairs), 49 contradictions were not symmetric. This is not as bad as humans did in the paper above, but it still shows a lack of consistency.

6.3 Entailment requires premises and hypothesis

The premise-only work in NLI points to the fact that the labels E, C, and N could be accurately determined without any premise, simply using the hypothesis. To make sure that our corpus does not have the same problem, we run an experiment using a dummy true premise, say, “Right adjoints preserve limits".

We substitute this sentence for the premise in all 279 pairs, and evaluate the new pairs using the Group 2 Models. These models do not suffer from the same problems that earlier ones did; all four essentially classified all of the hypotheses as N, which is correct.

7 Final remarks

We find it useful to discuss our work by seeing how it aligns with the perceptive conclusions drawn by (Madaan et al., 2024).[3] We agree that evaluating models on NLI tasks is still relevant. For Math NLI, we do not find models to be saturated. This contrasts with ordinary language NLI (ONLI). We also confirm their finding that “while the similarity of model distributions with human label distributions increases with scale, it is still much higher

---

[3] We would compare with other sources, but (Madaan et al., 2024) seems to be the most relevant contemporary paper on this topic.

		GPT4	L2	L3	C3	L3.1	Q2	Mixtral	DS	Ge2
C	precision	100.0	100.0	100.0	96.6	96.6	96.6	96.7	100.0	100.0
	recall	96.7	53.3	96.7	93.3	93.3	93.3	96.7	83.3	100.0
	f1-score	98.3	69.6	98.3	94.9	94.9	94.9	96.7	90.9	**100.0**
E	precision	70.7	57.7	66.7	71.8	65.2	75.0	76.3	75.0	73.0
	recall	96.7	100.0	100.0	93.3	100.0	100.0	96.7	90.0	90.0
	f1-score	81.7	73.2	80.0	81.2	78.9	**85.7**	85.3	81.8	80.6
N	precision	89.5	33.3	93.3	85.7	85.7	90.0	90.5	71.4	86.4
	recall	58.6	24.1	48.3	62.1	41.4	62.1	65.5	69.0	65.5
	f1-score	70.8	28.0	63.6	72.0	55.8	73.5	**76.0**	70.2	74.5
acc		84.3	59.6	82.0	83.1	78.7	85.4	**86.5**	80.9	85.4
avg	precision	86.7	64.0	86.6	84.7	82.5	87.2	87.8	82.3	86.4
	recall	84.3	59.6	82.0	83.1	78.7	85.4	86.5	80.9	85.4
	f1-score	83.8	57.2	80.8	82.8	76.8	84.8	**86.1**	81.1	85.2

Table 8: Results of LLMs on GPT-generated Corpus.

Model	E	N	C
L3.1	.039	.961	.0
Q2	.004	.992	.004
Mixtral	.0	1.00	.0
Ge2	.004	.996	.004

Table 9: Result of Hypothesis only Baseline

than the similarity between two populations of humans, making it a potentially interesting statistic to consider." We have found that models show less of a distribution of labels than humans. We mean that the models are closer to unanimity than humans. Finally, they note a certain "subjectivity": "examples with 'incorrect' predictions are rarely in fact incorrect; most concern questions on which humans may disagree as well." And just as they point out, "The ground truth labels for NLP benchmarks are often decided according to the majority label by human annotators. This simplifies the data annotation process while also making the evaluation easier. However, several previous studies have noted that human disagreements in annotations for NLP datasets reflect the lack of a single ground truth label, rather than noise in the annotation process." Even in mathematical texts, there is room for disagreements between experts.

7.1 Conclusion and future directions

This paper investigates the performance of Large Language Models (LLMs) on Natural Language Inference (NLI) tasks within the domain of research-level mathematics. We explore the complexities of mathematical language compared to everyday language and evaluate LLMs' ability to handle mathematical inferences, noting some surprising strengths and weaknesses.

Contrary to what we initially assumed Math NLI is not much easier than ONLI for LLMs. Challenges include unfamiliar vocabulary (e.g., 'ring', 'field', 'comonad'), multimodal elements like symbols and equations, lack of open-source mathematical resources, and the difficulty of finding expert human annotators.

LLMs show paradoxical performance on math tasks: despite exhibiting impressive capabilities in complex mathematical-like problem-solving, LLMs surprisingly struggle with basic logical reasoning and NLI tasks in mathematics. We have documented issues with self-consistency, which is crucial in mathematics. A sanity check testing whether LLMs correctly identify that a statement entails itself ($A \to A$) revealed a very small number of errors, but the explanations for these errors showed a fundamental reasoning flaw.

Post-GPT LLMs avoid some issues that plagued earlier systems. For example, we expected lexical ambiguity involving math words to cause LLMs to stumble, as in mixing up "stack" (a mathematical concept) with ordinary "stack" (pile). They did not do so.

We provide two corpora intended to support further research in the Math NLI area. One had hypotheses which we wrote ourselves, and the other had LLMs write the hypotheses. We believe that these corpora will help newcomers to this attractive area. And our results give some idea of what is reasonable to expect from this area in the next years.

Further directions include combining our work with theorem provers or other symbolic methods, tests of similarity as opposed to inference, and interactions of our work with running systems in the Math NLI area. We also leave to future work an analysis of the CoT explanations provided by LLMs. For us, this would be especially interesting regarding the red pairs (see Section 2.2).

Limitations

We did not fine-tune to mathematical text the LLMs we use. We also only ran things once. All of our mathematical work was centered on the relatively special area of category theory, since that was the source of our premise pairs. We do not expect significant differences when we pivot to other branches of mathematics.

A more problematic limitation is that from the outset we concentrated on a relatively limited kind of sentence. That is, we aimed for sentences which did not manifest interesting but semantically problematic phenomena like ellipses, temporal reference, poetic language, and the like. In a sense, we aimed for sentences that were close to what one could formalize in standard logic. This concentration was behind our initial choice of 432 sentences from the TAC corpus. We also wanted sentences which were not too short, not too long, and with little or no LaTeX. This also is a limitation.

Acknowledgments

The work of Hai Hu is funded by Shanghai Pujiang Program (22PJC063).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.

Jacob Collard, Valeria de Paiva, Brendan Fong, and Eswaran Subrahmanian. 2022. Extracting mathematical concepts from text. *Preprint*, arXiv:2208.13830.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Wesley H. Holliday, Matthew Mandelkern, and Cedegao E. Zhang. 2024. Conditional and modal reasoning in large language models. *Preprint*, arXiv:2401.17169.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Aikaterini-Lida Kalouli, Valeria de Paiva, and Livy Real. 2017. Correcting contradictions. In *Proceedings of the Computing Natural Language Inference Workshop*.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Preprint*, arXiv:2305.00050.

Lovish Madaan, David Esiobu, Pontus Stenetorp, Barbara Plank, and Dieuwke Hupkes. 2024. Lost in inference: Rediscovering the role of natural language inference for large language models. *arXiv preprint arXiv:2411.14103*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Anastasiia Sedova, Robert Litschko, Diego Frassinelli, Benjamin Roth, and Barbara Plank. 2024. To know or not to know? analyzing self-consistency of large language models under ambiguity. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17203–17217, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625:476 – 482.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2024. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *Preprint*, arXiv:2306.09841.

A On the LLMs used in this work

See Table 2. We used Qwen2-72B-Instruct, which was released in June 2024. According to the Qwen2 Technical Report, this model outperformed Llama3-70B-Instruct on most benchmarks, including mathematical benchmarks such as GSM8K and MATH.

B Disagreements between models and humans in the seed corpus

1. *P: Using these ideas, we also prove that magnetic monopoles form an abelian group.* H: Using these ideas, we also prove that monopoles form an abelian group.

   Humans say the label is N, as it's only for magnetic monopoles that we have the abelian group. Machines say entailment E, but no mathematician would state the weaker result, if they could prove it without the extra hypothesis.

2. *P: The problem of relating a factorization system to a pointed endofunctor is considered.* H: A pointed endofunctor cannot be related to a factorization system.

   Humans disagree: some say contradiction C, others say N

3. *P: This paper introduces the notions of vector field and flow on a general differentiable stack.* H: This paper generalizes the notions of vector field and flow on a stack.

4. *P: We define eventually cyclic Boolean flows and the eventually cyclic spectrum of a Boolean flow.* H: The definition of the eventually cyclic spectrum of a Boolean flow uses the definition of eventually cyclic Boolean flows.

5. *P: The axioms resemble those for monoidal Abelian categories with the addition of an involutive functor.* H: The axioms are the ones of monoidal Abelian categories.

6. *P: The category of Set-valued presheaves on a small category B is a topos.* H: The category of Set-valued presheaves on a small category C is a topos.

7. *P: The category of Set-valued presheaves on a small category B is a topos.* H: There exists a small category C such that the category of Set-valued presheaves on C is not a topos.

8. *P: Various concerns suggest looking for internal co-categories in categories with strong logical structure.* H: We suggest looking for internal co-categories.

C Seed corpus prompt

Here is the prompt which we used on the seed corpus:

`[Begin prompt head]`
Suppose you are a logician. Your job is to determine the inference relation between the premise and the hypothesis. There could be three answers: (1) the hypothesis is entailed by the premise; (2) the hypothesis is neutral to the premise; (3) the hypothesis contradicts the premise. Please first tell me your answer and explain why this is your answer.
`[End prompt head]`
Premise: `[Premise]`
Hypothesis: `[Hypothesis]`

All Entities are Not Created Equal: Examining the Long Tail for Ultra-Fine Entity Typing

Advait Deshmukh Ashwin Umadi Dananjay Srinivas Maria Leonor Pacheco
University of Colorado Boulder
`{advait.deshmukh, ashwin.umadi, dananjay.srinivas}@colorado.edu`

Abstract

Due to their capacity to acquire world knowledge from large corpora, pre-trained language models (PLMs) are extensively used in ultra-fine entity typing tasks where the space of labels is extremely large. In this work, we explore the limitations of the knowledge acquired by PLMs by proposing a novel heuristic to approximate the pre-training distribution of entities when the pre-training data is unknown. Then, we systematically demonstrate that entity-typing approaches that rely solely on the parametric knowledge of PLMs struggle significantly with entities at the long tail of the pre-training distribution, and that knowledge-infused approaches can account for some of these shortcomings. Our findings suggest that we need to go beyond PLMs to produce solutions that perform well for infrequent entities.

1 Introduction

Ultra-fine entity typing (UFET) is the task of inferring the type of an entity with high specificity (Choi et al., 2018). For example, in the sentence *"Barack Obama attended Biden's presidential inauguration."*, the entity "`Barack Obama`" would have a coarse entity-type of "`person`", but a more specific, ultra-fine entity type such as "`politician`", "`democrat`" or "`ex-president`". Previous work on UFET relies mostly on pre-trained language models (PLMs) to elicit entity types (Li et al., 2022, 2023; Dai et al., 2021; Pan et al., 2022). The common approach is to position the UFET task close to the learning objective of the PLM. For example, Pan et al. (2022) exploit the masked language modeling objective by appending an entity-mention and mask to a sentence, allowing BERT to fill in the mask to elicit a type (e.g., `Barack Obama attended Biden's presidential inauguration, Barack Obama is a [MASK]`). Such approaches are attractive because they benefit from the large amount of pre-training data the PLM has seen to make common associations, thus helping them determine the most likely type of entity based on the context provided.

Albeit successful, PLMs are limited by the data that they have been exposed to, so their reliability can be affected when working with infrequently used language constructions such as rarely/never seen entities. Since PLMs rarely encounter these entities, they have fewer opportunities to capture knowledge about them compared to frequent entities. Past work has mostly overlooked this issue, assuming that the embedding space will be enough to capture similarities between rare and less rare entities and generalize across them. Studies that have looked at infrequent entities usually characterize them in terms of their availability in task-specific training data (Choi et al., 2018; Schouten et al., 2022), rather than the data used to pre-train PLMs.

In this paper, we investigate the extent to which PLM-based methods struggle when determining the ultra-fine grained type of entities that occur infrequently in their pre-training data. This is a challenging undertaking, as the data used for pre-training PLMs are often unavailable (Shi et al., 2024). To address this challenge, we set out to answer the following research questions:

RQ1: Do internet search hits provide a good proxy for estimating relative entity frequency? To answer this question, we follow the conventional wisdom that modern PLMs have been trained on "all of the Internet", and we estimate the frequency of an entity by querying the Google search API to obtain the number of documents containing that entity. We validate this proxy by correlating it with real-world datasets known to be used in PLM pre-training, acknowledging that such disclosures are limited to only a few models.

RQ2: Do internal model representations reflect the entity frequencies in the pre-training data? To answer this question, we measure the correlation between the number of Google search

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 189–201
November 8-9, 2025 ©2025 Association for Computational Linguistics

hits obtained for an entity and the model's probability of eliciting that entity in numerous contexts. We perform this test across various PLMs, including both masked LMs and causal LMs, and find a strong correlation across the board.

RQ3: Do PLMs struggle to type entities that are in the long tail of the pre-training distribution? Finally, we design a benchmark to answer this question.[*] To do this, we divide entities in various UFET datasets into bins based on their resulting frequencies from our Google search. Our hypothesis is that the bins consisting of the least frequent entities "in the wild" will be much harder to predict than the rest. We test this assertion by looking at the performance of two types of published approaches for UFET; PLM-only approaches, which we expect to struggle at the long tail of the pre-training distribution, and knowledge-infused approaches, which we expect can level the playing field across frequencies by leveraging external sources of information. Our results confirm our hypothesis, suggesting that we need to go beyond PLMs to produce solutions that perform well for rare, new, or infrequent entities.

2 Related Work

UFET was proposed to generate free-form noun phrases that appropriately describe the type of a target entity (Choi et al., 2018). UFET has found several downstream applications, such as coreference resolution (Durrett and Klein, 2014), entity linking (Onoe and Durrett, 2020) and relation extraction (Yaghoobzadeh et al., 2017).

PLM solutions for UFET. PLMs have been shown to capture world knowledge in their parameter space (Roberts et al., 2020; Jiang et al., 2020). This ability has allowed them to perform well on 'fill-in-the-blank' problems, where the goal is to elicit an answer to a query based on the context provided. Many UFET approaches have capitalized on this ability; Dai et al. (2021) looked at introducing Hearst patterns with `[MASK]` tokens to describe entity types using BERT, while Pan et al. (2022) primed BERT to produce an ultra-fine type by appending the entity mention and a `[MASK]` token to a sentence. While these methods report some of the best metrics for UFET, they do not explore how their approaches fare on infrequent, rare entities that PLMs may not have been sufficiently exposed to. In this work, we explore the effect of entity representation strength on a variety of PLMs in order to determine whether rarer entities cause significant issues for PLM-based approaches.

Knowledge Infused UFET. The UFET dataset has 9 general, 121 fine, and $10,201$ ultra-fine types. While a large type vocabulary and the scarcity of annotated examples per type make this task especially challenging, type labels often consist of rich semantics. Li et al. (2022) leverage the type semantics and formulate the task as an NLI problem. Others have exploited the dependencies between labels (Liu et al., 2021) and hierarchies within types (Onoe et al., 2021) to supplement the PLM objectives. Such techniques often perform significantly better than PLM-only approaches.

Approaches to Estimate Entity Frequencies in Large Corpora. While there have been efforts to efficiently index large pre-training corpora to better estimate entity frequencies (Liu et al., 2024; Xu et al., 2025), we argue that there are two clear limitations with this approach: (1) the assumption that training data for models is always available and (2) the need to re-index datasets as they are updated over time. Therefore, we need simple proxies that can approximate the entity distribution "seen" by PLMs without direct access to their training data.

3 Experimental Design

Our study is composed of three experiments. First, we evaluate a proxy to establish the relative frequency of entities in pre-training data using Search Engine API indexing. Then, we examine whether parametric representations match these frequencies by correlating the likelihood of a PLM suggesting a target entity to how the term is indexed on the Web. Lastly, we select seven models to perform ultra-fine entity typing; three based on PLM objectives, three that leverage some additional knowledge, and one simple baseline. We measure typing performance across entity groups of varying frequency across the Internet data.

Establishing the Long Tail (RQ1, RQ2) The long tail of an entity distribution is characterized by a large number of entities that are rare to encounter in the real world. Assuming that the Internet is a fairly balanced representation of a real-world distribution, we use the Internet search hits as a proxy for frequency. We quantify the occurrence "in the wild" of each entity in our dataset by performing strict searches using the Google Search API.

---

[*] Data and Code available at: `https://github.com/blast-cu/All-Entities-are-Not-Created-Equal`

Bin	# of Examples	Representative Entity	Avg. # of Tokens
1	301	the Baton Rouge police chief and the serial murder task force	11.63
2	301	Left fielder Carl Crawford	4.35
3	300	The Polish government	2.67
4	1095	the film	1.18

Table 1: Distribution of entities across UFET test bins

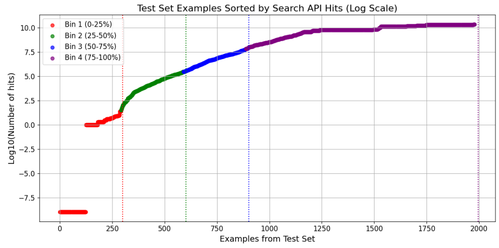


Figure 1: Entity distribution across **UFET** test bins

As most datasets used to pre-train modern PLMs come from a subset of the Internet, we hypothesize that such "long-tail" entities would be underrepresented in the parameter space of these models. To test this hypothesis, we query models and assess whether an entity is underrepresented based on its ability to be predicted or reconstructed accurately. We then measure the correlation between Internet frequencies and model-estimated probabilities.

To perform model estimations reliably, we must balance different considerations. First, given that PLMs learn word representations from in-context examples, we need a representative set of in-context examples for each of the entities that we want to estimate. To do this, we prompt *Llama3-8B-Instruct* to generate 10 different sentences that include the target entity. Second, we need to consider the training objective of the PLM when querying it for the probability of a given entity in a given context. To recover entity probabilities with <u>Masked LMs</u>, we first replace entity tokens with `[MASK]` tokens. Then, we generate a probability distribution over all candidate tokens. To deal with <u>Causal LMs</u>, we reframe the task as a *fill-in-the-blank* problem. We provide the model with a prompt (see App E) and compute the probabilities through a conditional generation process.

In both cases, we use the probability that the model assigns to the tokens of the target entity to represent its salience in the model's parametric knowledge. More details about the probability estimation process can be found in App. D.

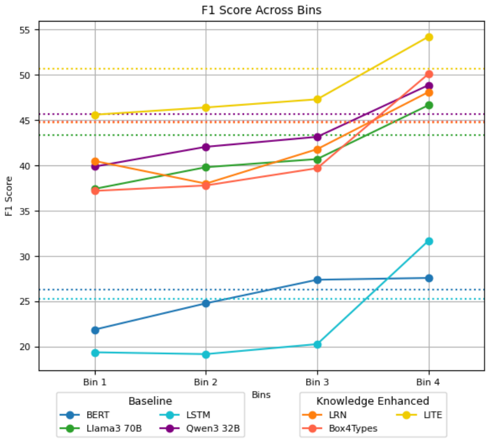
Measuring Impact on Typing (RQ3) We examine how the real-world distribution of entities affects the performance of entity-typing models. To define rare entities, we calculate frequency scores for all target UFET entities using the Google Custom Search API and group them into four bins based on quartiles. Bin 1 includes the rarest entities, while Bin 4 contains the most frequent; based on their occurrence on the Internet. Representative examples are presented in Tab. 1. The entity distribution across the bins is visualized in Fig. 1.

We select seven representative models to test against the bin splits. The models can be roughly divided into two categories. (1) Naive <u>PLM-based</u> approaches that rely on PLM objectives such as MLM (BERT family of models) or Causal LM (Llama3 and Qwen3) to predict types. We include implementation details in App. G. (2) <u>Knowledge-infused</u> approaches that exploit additional information embedded within or relevant to the type labels. LITE (Li et al., 2022) formulates entity typing as an NLI problem. LRN (Liu et al., 2021) exploits intrinsic and extrinsic dependencies between label types. Box4Types (Onoe et al., 2021) relies on box embeddings to capture hierarchies of types. More details about these systems can be found in App G. For completeness, we also benchmark an LSTM model. This is a supervised model built on top of pre-trained word embeddings, which consider similar representation objectives to those of PLMs (word co-occurrence).

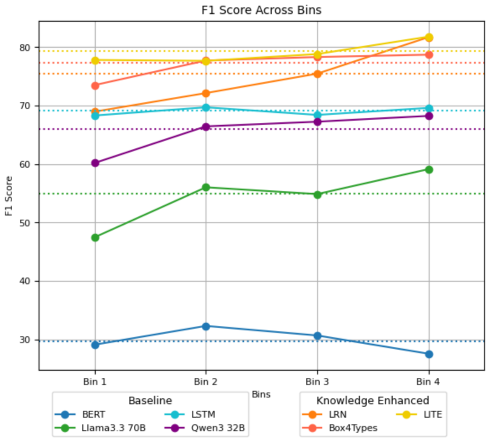
We use the crowd-annotated portion of the UFET dataset (Choi et al., 2018) for our experiments. This dataset contains entity mentions with their surrounding context and the ultra-fine types associated with them. The dataset of 5,994 tuples is divided into train/test/dev splits, each containing 1,998 tuples. We use OntoNotes (Gillick et al., 2016) as a secondary dataset, which has a train/dev/test split of 250k/2k/9k examples.

4 Evaluation and Discussion

Long Tail Analysis (RQ1, RQ2) We compare average PLM probability estimates with the number of hits the Search Engine API has for the target entity. We do this for three PLMs with both MLM and Causal LM learning objectives: BERT (Devlin et al., 2019), BART (Lewis et al., 2020) and Qwen3 (Yang et al., 2025). We calculate the probability assigned by the PLMs per in-context example



(a) UFET



(b) OntoNotes

Figure 2: Baseline vs. Knowledge-enhanced Performance across test bins

using the method described in Section 3, and take the average across all sentences for its respective entity. We perform a correlation analysis between PLM probability estimates and the entity frequencies obtained from the Google Search API and observe high Spearman correlation coefficients (0.885 for BERT, 0.716 for BART and 0.897 for Qwen3). To visualize this, we plot the hits from the Search Engine API against the average probability for the entity obtained by each model in App. F. The high correlation between the PLM probability estimates and the number of API hits supports our hypothesis: entities that occur more/less frequently in the real world are more/less salient in PLMs.

Temporal shifts. We recognize that the Internet is constantly evolving and that temporal dynamics could potentially alter the distribution of entities. For this reason, we performed our analysis with API data capped at 2018 and at 2024, and found the results to be consistent over time, with minor changes in correlation coefficients (See App. B).

Training Corpus	Spearman Correlation
BookCorpus	0.583
C4-train	0.957
Pile-train	0.954
RedPajama	0.957
Dolma-v1.7	0.961

Table 2: Spearman correlation coefficient of **UFET** entity frequencies as estimated by search API vs counts from actual pre-training corpora

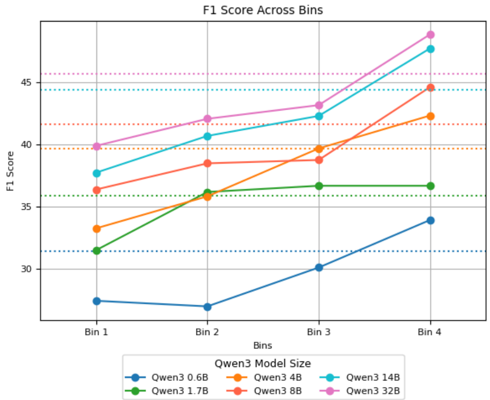
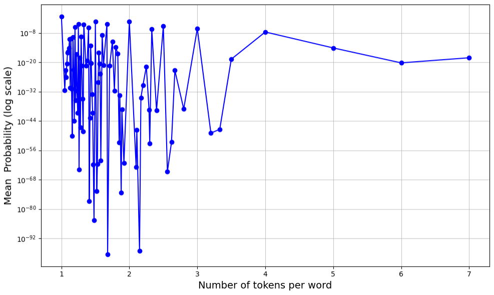


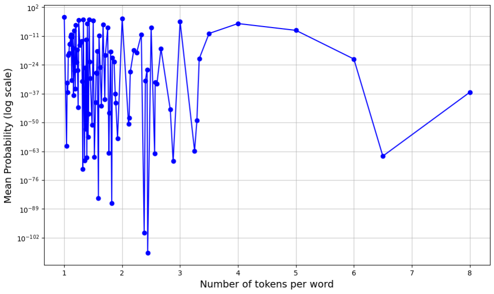
Figure 3: Effect of scaling on performance across **UFET** bins

Search API vs. real pre-training datasets. We test the validity of using Internet counts to estimate the pre-training distribution by performing a correlation study between search API hits and real-world pre-training datasets such as BookCorpus (Zhu et al., 2015), C4 (Raffel et al., 2020), Pile (Gao et al., 2020; Biderman et al., 2022), RedPajama (Weber et al., 2024), Dolma-v1.7 (Soldaini et al., 2024). We estimate the entity counts for BookCorpus by performing strict searches. For Pile, we rely on the Infini-gram-mini hosted API (Xu et al., 2025). For the remaining datasets (C4, RedPajama, Dolma); we use the Infini-gram hosted API (Liu et al., 2024). N-gram counts in Infini-gram and Infini-gram-mini are case-sensitive and therefore noisy. Regardless, we see high spearman coefficients (>0.9) for all of them. While the correlation coefficient for BookCorpus is significantly lower, we find that the largest contributing factor is its much smaller scale. As a result, many entities are not present in BookCorpus. When restricting only to entities that are present in BookCorpus, we find that the correlation jumps as high as 0.88.

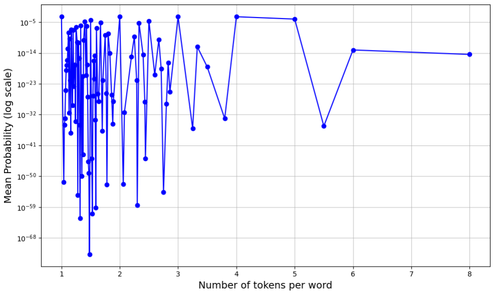
For completeness, we also report the Spearman correlation coefficients between the frequency of UFET entities in different pre-training datasets and the average entity recovery probability given surrounding context (see App. A).



(a) BERT-Base-Uncased



(b) BART-Large



(c) Qwen3-0.6B

Figure 4: Average **UFET** entity recovery probability versus average number of tokens per word for three model tokenizers

Typing Performance (RQ3) We benchmark the seven models introduced in Sec. 3: UFET-LSTM (Choi et al., 2018), few-shot Llama3 (Dubey et al., 2024)/Qwen3 (Yang et al., 2025), BERT (PLM-based), and LITE (Li et al., 2022), LRN (Liu et al., 2021) and Box4Types (Onoe et al., 2021) (Knowledge-infused), and plot results in Fig. 2.

We find that most of the examined approaches perform better in determining the ultra-fine entity type when the target entity is more frequent (Bin4), and worse when the entity is rarer (Bin1). We note that most PLM approaches show a major decline in performance as we go down the bins (or underperforming completely, as in the case of BERT for OntoNotes). This performance decline is most underlined when moving from Bin 4 to Bin 3. Looking at the knowledge-infused approaches, we find that the enrichment from auxiliary tasks or label dependencies helps overcome the overreliance on learned entity representations for Bin 1. LITE performs the best, beating the rest by achieving an $F1$ score of 45.6 on infrequent entities and 54.2 on frequent entities for UFET, and an $F1$ score of 77.8 on infrequent entities and 81.8 on frequent entities for OntoNotes. This further highlights the advantage of the auxiliary task (NLI), which allows the model to transfer more knowledge than approaches that only rely on the PLMs parameters.

Effects of tokenizers. If a specific word is unknown to a PLM, it follows that the tokenizer will split it into multiple tokens. For this reason, we investigate whether words being split into multiple tokens causes a degradation in the probability of recovering an entity by looking at the splitting ratio (number of tokens / number of words). While we see a marked difference between a ratio of 1 and a ratio $\neq 1$, our results do not suggest any incremental effect afterwards (see Fig. 4). This suggests that our API Search proxy is a better, more nuanced indicator of entity rarity than the PLM tokenizer.

Other levels of granularity. We look at performance across frequency bins at varying entity type granularity (i.e., fine, coarse) and find the same trend of performance decay across buckets within each granularity level (see App. H).

Effect of Scaling. To understand how the size of the LM affects the typing task across bins, we evaluate Qwen3 models of different sizes (Fig. 3). We chose Qwen3 given the availability of a large range of model sizes of the same family. We find that regardless of model size, the effect of entity frequency in performance degradation is still observed. We include results for Llama3 in App. G.

5 Conclusion and Future Work

We showed the effectiveness of internet search API hits as a proxy for entity frequency in large pretraining corpora (RQ1). We also showed that, as expected, this distribution significantly affects entity representations for different PLMs (RQ2). Finally, we measured the performance of several PLM-based and knowledge-infused entity typing systems on entities with different frequencies and found that all models performed worse on rarer/less-probable entities (RQ3). However, we found that all knowledge-infused systems are considerably more robust to frequency shifts than PLM-based approaches. Our results show that for PLMs to fulfill their promise for long-tail entity typing, we need better strategies to inject knowledge about rare entities into PLMs by using external resources and other forms of domain knowledge.

Limitations

Due to limited resources, we used quantized versions of the models when studying the impact of scaling on entity typing systems in Sec. 4. Previous work has shown that techniques such as quantization can impact the performance of LLMs, especially in low-resource settings (Diddee et al., 2022). We acknowledge that quantization may be further exacerbating the general struggle that PLMs have with entities at the long-tail of the distribution.

Ethical Considerations

To the best of our knowledge, this work does not incur any violation of the code of ethics. We used models that were Open Sourced by their authors, with code available online. All the information required to replicate our experiment is provided in the paper. We use Llama3/Qwen3, large language models whose weights may be updated by the model's creator. In such a case, we caution that some results may not be reproducible exactly, but believe that our findings will still hold.

In the interest of space, we moved some plots and details to the appendix.

References

Stella Biderman, Kieran Bicheno, and Leo Gao. 2022. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1790–1799, Online. Association for Computational Linguistics.

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

Harshita Diddee, Sandipan Dandapat, Monojit Choudhury, Tanuja Ganu, and Kalika Bali. 2022. Too brittle to touch: Comparing the stability of quantization and distillation towards developing low-resource MT models. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 870–885, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2016. Context-dependent fine-grained entity type tagging. *Preprint*, arXiv:1412.1820.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622.

Na Li, Zied Bouraoui, and Steven Schockaert. 2023. Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy. *Preprint*, arXiv:2305.12802.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021. Fine-grained entity typing via label reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4611–4622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8576–8583.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4317–4323. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stefan Schouten, Peter Bloem, and Piek Vossen. 2022. Probing the representations of named entities in transformer-based language models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 384–393, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 116462–116492. Curran Associates, Inc.

Hao Xu, Jiacheng Liu, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. Infini-gram mini: Exact n-gram search at the internet scale with fm-index. *Preprint*, arXiv:2506.12229.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Analysis of UFET entities in pre-training datasets

We report the spearman correlation values for UFET Entity counts as obtained in pre-training datasets and their recovery probabilities in Tab. 2.

B Temporal dynamics of search API hits

We use the Google Search API to approximate the distribution of entity frequencies that models have seen during training. While convenient, this approach may potentially ignore the temporal changes that might occur in the distributions of such entities. This is especially important as models we discuss in our work (BERT, BART and Qwen3) have been trained at different points within the last decade. To ensure that these dynamics do

Model	BookCorpus	Pile-train	C4-train	Dolma-v1.7	RedPajama
Bert-base-uncased	0.571	0.883	0.886	0.891	0.886
Bart-large	0.359	0.705	0.707	0.715	0.708
Qwen3-0.6B	0.514	0.895	0.897	0.902	0.901

Table 3: Spearman correlation coefficient of **UFET** entity frequencies in pretrained datasets vs Average entity recovery probability given surrounding context

Model	API till 2018	API till 2024
Bert-base-uncased	0.883	0.885
Bart-large	0.711	0.716
Qwen3-0.6B	0.895	0.897

Table 4: Spearman correlation coefficient of API entity frequencies vs Average **UFET** entity recovery probability given surrounding context (API results capped at 2018 and 2024)

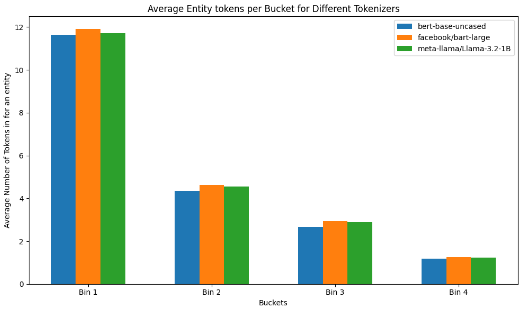


Figure 5: Average number of tokens for **UFET** test bins

not significantly impact our claims, we compare the correlation between the LM predictions and the API data capped at 2018 and 2024. We find results to be largely consistent across these two time periods (See Tab. 4). On further examination, we find that between 2018 and 2024 only 39 entities from the test set (<2%) change their bin classification. For our main results, we rank entities using the 2024 results.

C UFET test bin distribution

To better visualize the distribution of entities across bins, we plot the log of API hits in Fig. 1. We also compare the average number of tokens as obtained with different tokenizers for each of our test bins in Fig. 5.

D Method for Generating Masks and Calculating Entity Probability

An entity can be comprised of a single token or a multi-token phrase. For multi-token entities we employ a conditional generation approach where we generate the entity sequentially, one token at a time, moving from left to right. The probability of the entire entity is determined by product of the conditional probability of each token t_i conditioned on all preceding tokens and the surrounding context C, where i represents the position of the token in the sequence and n is the total number of tokens comprising the entity.

$$P(t_1, t_2, \ldots, t_n \mid C) = \prod_{i=1}^{n} P\left(t_i \,\middle|\, t_1, t_2, \ldots, t_{i-1}, C\right) \tag{1}$$

We use this approach to benchmark three models with comparable sizes and distinct pre-training objectives.

BERT (MLM): To recover an entity using BERT (bert-base-uncased), we first replace the entity with an equal number of `[MASK]` tokens. Then we calculate the probability the entity being recovered as described before, replacing one `[MASK]` per iteration.

BART (MLM): For BART (bart-large) (Lewis et al., 2020), we take a similar approach but with only a single `<mask>` token. We progressively expand the `<mask>`, one token at a time, calculating the probability for each subsequent token until the entire entity is recovered.

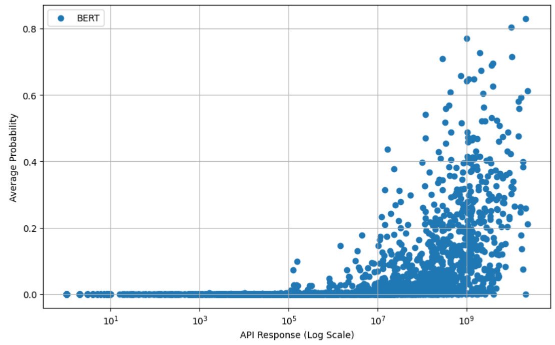
Qwen3 (Causal LM): Since Qwen3 (Yang et al., 2025) is not pre-trained with an MLM objective, we reframe the task as a *Fill-in-the-Blank* problem using *Qwen3-0.6B*. We provide the model with a prompt (See App. E) and compute the probabilities through a conditional generation process, one token at a time.

E Prompt used to calculate the probability to recover an entity for Qwen3

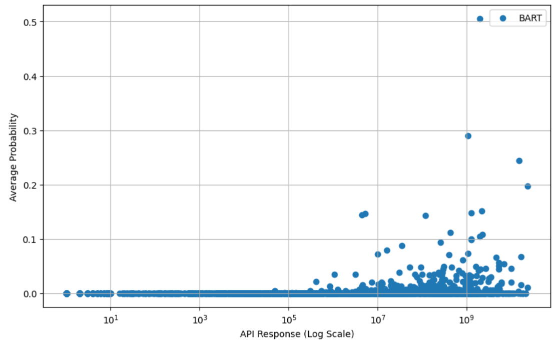
The prompt used to calculate the probability to recover an entity for Qwen3 is:

`Instruction:` Fill in the appropriate entity that completes the sentence below.

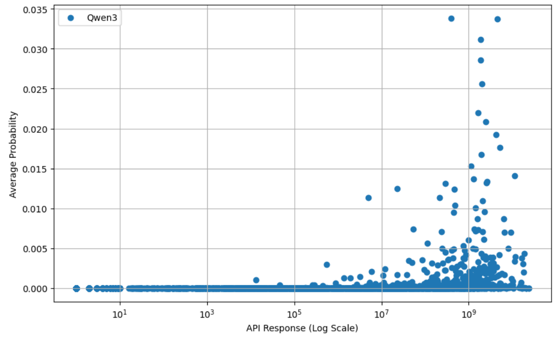
`Context:` {sentence with the entity mention replaced by a [blank]}



(a) BERT-Base-Uncased: Average probability vs. Log search-engine hits



(b) BART-Large: Average probability vs. Log search-engine hits



(c) Qwen3-0.6B: Average probability vs. Log search-engine hits

Figure 6: Average **UFET** Entity Recovery Probability vs Search API Hits

Response: `[blank]` can be replaced with:

F Scatter plots of Entity Recovery Probability in BERT, BART and Qwen3 against Search Engine API hits

To visualize the high Spearman correlation coefficient between entity recovery probabilities and search engine API hits, we plot Fig. 6.

G Models for entity typing

G.1 BERT - Baseline

Inspired by (Dai et al., 2021), we frame the typing problem as mask prediction task for BERT (bert-base-uncased). We use hearst-like patterns ("[MASK] such as entity mention", "entity mention and any other [MASK]", "entity mention and some other [MASK]") and conduct experiments to find the optimal templates. The top 'n' predictions for the [MASK] (with plural → singular conversion, restricted to the type vocabulary) are considered as candidate labels. we find the optimal value for 'n' (number of labels) by experimenting with the development set.

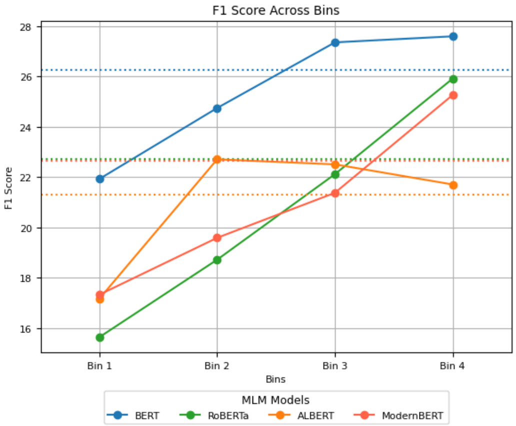


Figure 7: Evaluation of MLM models across **UFET** test bins

`{Hearst}` n = 12, F1 = 0.0661

`{Sentence} . {Hearst}` n = 5, F1 = 0.2277

`{Sentence} [SEP] {Hearst}` n = 5, F1 = 0.2338

`{Hearst} inserted in {sentence}` n = 6, F1 = 0.2631

We experiment with different MLM models (See Fig. 7) and present the results from the best performing setting in Tab. 5. We find that the trend is largely consistent among them, with the exception of ALBERT which converges by Bin 3.

G.2 Llama3/Qwen3 - Baseline

We model the entity typing problem as a few-shot task for *Llama3* and *Qwen3* models to evaluate its efficacy in entity typing. We experiment with the number of examples (from the train set) in the prompts in increments of five examples. We found that the performance was optimal for 15-examples in the prompt and used that setting for the rest of our experiments.

We use the following system prompt:

```
# Entity-Typing Assistant
You are a precise entity-typing assistant.
```

```
Given a sentence in which **one entity mention
    ↪ is wrapped in ‘<ENT> ... </ENT>‘ tags**,
    ↪ produce **only** a JSON object whose
    ↪ single key is **"predicted_types"**.

## Guidelines
- The value must be a JSON array of strings.
- Include all the type labels that are relevant.
- Remove duplicates and keep each type concise
    ↪ (ideally a short noun phrase).
- Do not output any keys other than
    ↪ ‘"predicted_types"‘.

## Input Format
- SENTENCE: The complete sentence with the
    ↪ target entity clearly marked with
    ↪ ‘<ENT>‘ tags
- ENTITY_MENTION: The target entity mention
    ↪ from the sentence

## Output Format
‘‘‘json
{
    "predicted_types": ["TypeA", "TypeB",
    ↪ "TypeC", ...]
}
‘‘‘
```

Followed by examples from the train set in this format:

```
# Example #{i}:
- INPUT:
- SENTENCE: '{sentence}'
- ENTITY_MENTION: '{entity_mention}'

- OUTPUT:
{{\"predicted_types\": [{types}]}}
```

With the input prompt as specified above, we generate the response from the model with the json schema for generation passed to the model (format parameter in ollama's client.chat method). Passing the generation schema ensures that the model adheres to the expected format and prevents malformed/incorrect json output. We filter out the generated types to match the type vocabulary for the respective dataset.

We also studied the effect of scaling on typing performance. Specifically, it was of interest to us to understand if scaling can help bridge performance discrepancies between Bin 1 and Bin 4. Although we did not find significant trends, the gains in performance made by the models generally seemingly favor buckets with higher entity frequency, exacerbating the long-tail problem. (see Fig. 3 and Fig. 8)

Note: all the models we evaluate are the 4 bit quantized models (q4_K_M) available through Ollama.

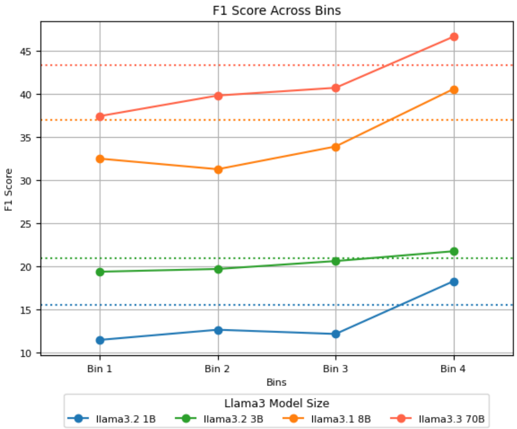


Figure 8: Effect of scaling on model performance across **UFET** test bins

G.3 UFET-LSTM (Choi et al., 2018)

UFET-LSTM frames mention typing as predicting free-form labels (e.g., criminal, victim). Because the label set is so large and unconstrained, they adopt a neural model that represents both the mention and its context: a BiLSTM with attention for the sentence, plus a CNN with attention for the mention span. Although the authors' reported results rely on distant supervision and crowd sourced data, our analysis uses only the crowd sourced training data from UFET. Accordingly, we train a new LSTM-based model using that dataset and present the per-bin performance in Tab. 8.

G.4 LITE (Li et al., 2022)

LITE approaches the ultra-fine entity typing by reframing it as an NLI task: treat the original sentence (with the target entity) as a premise, and generate short textual descriptions of the entity as hypotheses using a predefined structure. A pretrained NLI model scores how strongly each description is entailed. Using a learning-to-rank objective, LITE distinguishes correct types from incorrect ones. For our experiments to find the performance across bins, we use the final results as shared by the authors on the test dataset of crowd sourced UFET. Table 11 uses these predictions while capturing its performance. For OntoNotes, we recreated the model to the best of our ability.

G.5 Box4Types (Onoe et al., 2021)

Box4Types deploys box embeddings to effectively capture the hierarchies of types. The model represents both types and entity mentions as boxes. Each mention and its context are fed into a BERT-based model to embed that mention in our box space; essentially, the model leverages typological clues present in the surface text to hypothesize a type representation for the mention. This helps the model capture latent hierarchies better than the vector-based counterparts. We recreated the original approach described in paper and evaluated it against our bins in Tab. 10.

G.6 Label Reasoning Network (Liu et al., 2021)

Label Reasoning Network sequentially reasons about fine-grained entity labels by discovering and exploiting knowledge about label dependencies entailed by the data. These implicitly and explicitly entailed dependencies provide critical information which help the model overcome limitations of baseline LM approaches. The BERT based model leverages deductive and inductive reasoning. We recreated the models (without retrieval) as described in the paper and reported the results against our bins in Tab. 9.

H Fine grained evaluation of the models studied

We look at the performance of the discussed models across bins and label granularities (Coarse, Fine, Ultra-fine) as first proposed by (Choi et al., 2018).

The trend of decline in performance between Bin 4 to Bin 1 continues into the fine grained evaluation for the models. For each level of label granularity we find clear separation in performance levels as we move between the bins, suggesting that binning provides a measure of difficulty independent of the label granularity. This provides a unique opportunity to approach the entity typing task from a new perspective.

Table 5: BERT MLM (bert-base-uncased)

	Overall			Coarse			Fine			Ultra-fine		
Subset	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	23.7	29.5	26.3	64.9	43.0	51.7	34.2	40.7	37.1	16.8	22.5	19.2
Bin 1	19.2	25.5	21.9	59.0	41.8	48.9	22.4	34.3	27.1	14.1	19.0	16.2
Bin 2	22.5	27.7	24.8	57.6	34.2	43.0	31.7	43.2	36.6	17.3	23.2	19.8
Bin 3	24.6	31.0	27.4	60.1	41.0	48.7	34.4	42.1	37.9	18.8	25.1	21.5
Bin 4	25.1	30.8	27.6	69.1	45.8	55.1	38.2	41.1	39.6	16.8	22.5	19.2

Table 6: Llama3.3-70B

	Overall			Coarse			Fine			Ultra-fine		
Subset	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)
Full Test	45.9 ± 0.1	41.1 ± 0.1	43.4 ± 0.1	78.2 ± 0.3	68.7 ± 0.2	73.1 ± 0.3	58.0 ± 0.3	56.0 ± 0.2	57.0 ± 0.2	36.6 ± 0.2	30.0 ± 0.2	33.0 ± 0.2
Bin 1	38.3 ± 0.3	36.6 ± 0.7	37.4 ± 0.5	70.7 ± 0.5	54.8 ± 0.7	61.7 ± 0.4	48.5 ± 0.7	44.4 ± 1.0	46.4 ± 0.9	32.1 ± 0.3	30.5 ± 0.7	31.3 ± 0.5
Bin 2	43.2 ± 0.3	36.9 ± 0.3	39.8 ± 0.2	73.7 ± 0.8	50.9 ± 0.9	60.2 ± 0.9	55.0 ± 1.0	54.8 ± 0.8	54.9 ± 0.5	36.5 ± 0.7	30.0 ± 0.3	33.0 ± 0.4
Bin 3	44.4 ± 0.2	37.6 ± 0.3	40.7 ± 0.2	75.7 ± 1.0	59.5 ± 1.1	66.7 ± 1.1	59.4 ± 1.3	53.0 ± 0.9	56.0 ± 1.0	34.7 ± 0.4	28.4 ± 0.2	31.2 ± 0.2
Bin 4	49.0 ± 0.2	44.5 ± 0.1	46.6 ± 0.1	80.9 ± 0.3	78.2 ± 0.5	79.5 ± 0.4	60.4 ± 0.1	59.7 ± 0.4	60.1 ± 0.2	38.4 ± 0.2	30.4 ± 0.3	33.9 ± 0.3

Table 7: Qwen3-32B

	Overall			Coarse			Fine			Ultra-fine		
Subset	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)	P(±σ)	R(±σ)	F1(±σ)
Full Test	50.6 ± 0.6	41.6 ± 0.6	45.7 ± 0.3	77.6 ± 0.5	75.1 ± 0.9	76.4 ± 0.3	57.6 ± 0.8	55.9 ± 0.9	56.7 ± 0.4	39.5 ± 0.5	29.0 ± 0.5	33.5 ± 0.4
Bin 1	41.1 ± 1.0	38.7 ± 0.8	39.9 ± 0.7	67.0 ± 0.9	67.5 ± 0.8	67.2 ± 0.8	48.0 ± 1.7	48.5 ± 1.5	48.2 ± 1.4	33.2 ± 0.7	29.5 ± 0.9	31.3 ± 0.7
Bin 2	46.1 ± 0.6	38.7 ± 0.6	42.1 ± 0.3	74.6 ± 1.4	66.6 ± 1.1	70.4 ± 0.8	52.5 ± 0.7	53.9 ± 1.9	53.2 ± 0.7	37.8 ± 0.8	28.6 ± 0.3	32.6 ± 0.4
Bin 3	48.4 ± 1.1	39.0 ± 0.6	43.2 ± 0.4	71.7 ± 1.8	68.7 ± 0.6	70.1 ± 0.6	57.6 ± 1.2	54.6 ± 1.4	56.1 ± 1.2	38.5 ± 1.0	28.4 ± 1.0	32.7 ± 0.8
Bin 4	55.0 ± 0.6	44.0 ± 0.7	48.9 ± 0.3	82.2 ± 0.4	80.2 ± 1.0	81.2 ± 0.5	61.2 ± 0.9	58.3 ± 1.4	59.7 ± 0.7	42.1 ± 0.6	29.1 ± 0.7	34.4 ± 0.4

Table 8: LSTM

	Overall			Coarse			Fine			Ultra-fine		
Subset	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full test	41.7	18.2	25.3	57.3	52.8	55.0	41.7	16.2	23.4	27.3	7.7	12.0
Bin 1	33.4	13.7	19.4	42.3	39.9	41.0	18.2	8.1	11.2	24.4	5.7	9.2
Bin 2	34.3	13.3	19.2	42.4	38.4	40.3	45.6	19.6	27.4	29.2	7.0	11.4
Bin 3	32.6	14.7	20.3	46.4	40.9	43.5	43.8	15.4	22.8	25.0	7.2	11.1
Bin 4	47.5	23.8	31.7	69.4	63.9	66.6	45.6	21.0	28.8	28.0	10.5	15.3

Table 9: Label Reasoning Network

	Overall			Coarse			Fine			Ultra-fine		
Subset	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	57.1	33.7	42.4	75.7	76.6	76.1	57.5	46.8	51.6	44.0	19.8	27.3
Bin 1	52.7	27.4	36.0	65.8	69.5	67.6	52.8	42.1	46.9	43.4	14.5	21.8
Bin 2	49.2	26.8	34.7	65.6	67.7	66.7	47.4	38.8	42.7	37.1	15.7	22.0
Bin 3	52.3	27.9	36.4	67.3	68.7	68.0	55.0	37.6	44.7	38.8	17.3	23.9
Bin 4	61.5	38.9	47.7	82.6	82.0	82.3	61.4	52.3	56.5	47.0	23.2	31.1

Table 10: Box4Types

Subset	Overall			Coarse			Fine			Ultra-fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	52.8	38.9	44.8	70.5	82.9	76.2	52.9	53.4	53.2	45.4	24.5	31.8
Bin 1	46.3	31.0	37.2	58.9	77.7	67.0	42.9	43.6	43.2	38.8	16.0	22.7
Bin 2	47.7	31.4	37.8	60.0	76.7	67.3	45.3	46.3	45.8	44.5	20.5	28.0
Bin 3	47.8	33.9	39.7	60.2	79.9	68.7	49.6	53.8	51.6	42.8	20.6	27.8
Bin 4	57.4	44.4	50.1	79.4	86.2	82.6	58.3	57.3	57.8	47.7	29.0	36.1

Table 11: LITE

	Overall			Coarse			Fine			Ultra-fine		
Subset	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	54.8	47.1	50.7	74.5	81.7	77.9	61.4	57.2	59.3	44.3	35.5	39.4
Bin 1	50.7	41.4	45.6	64.7	76.3	70.0	53.8	42.4	47.4	41.0	31.8	35.8
Bin 2	49.7	43.5	46.4	65.7	76.0	70.4	61.1	56.2	58.6	41.9	34.1	37.6
Bin 3	49.2	45.6	47.3	63.9	73.9	68.6	57.1	53.7	55.4	43.3	36.3	39.5
Bin 4	58.8	50.2	54.2	81.7	86.0	83.8	64.2	61.7	62.9	46.1	36.7	40.8

When Does Meaning Backfire? Investigating the Role of AMRs in NLI

Junghyun Min[☆] **Xiulin Yang**[☆] **Shira Wein**[◇]
[☆]Georgetown University [◇]Amherst College
`{jm3743, xy236}@georgetown.edu` `swein@amherst.edu`

Abstract

Natural Language Inference (NLI) relies heavily on adequately parsing the semantic content of the premise and hypothesis. In this work, we investigate whether adding semantic information in the form of an Abstract Meaning Representation (AMR) helps pretrained language models better generalize in NLI. Our experiments[1] integrating AMR into NLI in both fine-tuning and prompting settings show that the presence of AMR in fine-tuning hinders model generalization while prompting with AMR leads to slight gains in `GPT-4o`. However, an ablation study reveals that the improvement comes from amplifying surface-level differences rather than aiding semantic reasoning. This amplification can mislead models to predict non-entailment even when the core meaning is preserved.

1 Introduction

Since the advent of large language models (LLMs), there has been ongoing debate about the utility of symbolic representations such as Abstract Meaning Representations (AMRs; Banarescu et al., 2013) in (LLM-based) pipelines and existing NLP tasks. While some studies report limited or negative impact of AMRs on mainstream NLP tasks (Jin et al., 2024), recent work has demonstrated their value in specific applications, such as syntactic simplification (Yao et al., 2024) and semantically-controllable text transformation (Li et al., 2025). Perhaps unsurprisingly, incorporating AMR has been particularly well-explored and effective in tasks related to semantics (Wein and Opitz, 2024).

Natural language inference (NLI; Dagan et al., 2010) is a popular task in NLP where the solver is given a *premise* and a *hypothesis*, and asked to determine whether the hypothesis is true if the premise is true. The label space consists of three labels: **entailment** if the hypothesis is true, **contradiction** if the hypothesis is false, and **neutral** if the truth value of the hypothesis cannot be determined; this can also be condensed in two labels: entailment and non-entailment. As shown in Figure 1 “Athletes introduced the secretaries” should be entailed by “Serious athletes introduced the secretaries.” Therefore, the label should be **entailment** because the truth of the premise indicates truth of (or *entails*) the hypothesis.

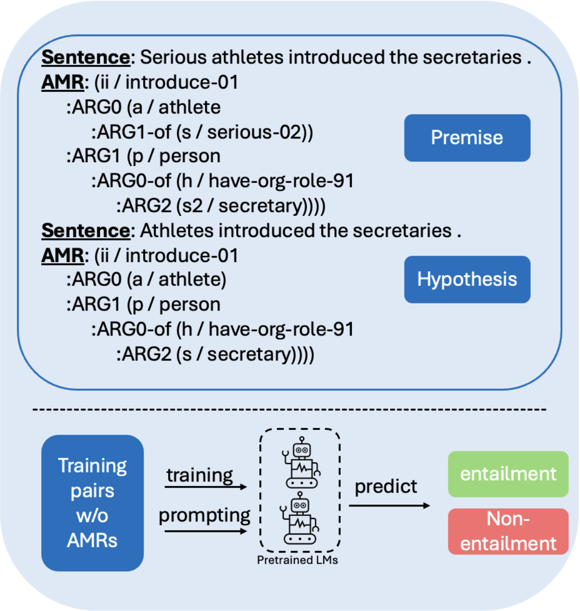


Figure 1: An example of NLI and experiment pipeline of the paper. AMRs are shown in penman notation.

As a meaning-focused task, NLI aligns well with the motivation behind AMRs, i.e., to abstract sentence meaning beyond surface form, given NLI models’ tendencies to adopt shallow heuristics rather than understanding the relationship between the premise and the hypothesis, leading to poor generalization to novel data (Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019; Serrano et al., 2023). In this paper, we inves-

---

[1] We publicly release our code at `https://github.com/Aatlantise/advarsarial-nli-amr`.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 202–211
November 8-9, 2025 ©2025 Association for Computational Linguistics

tigate whether incorporating AMRs as additional input - either during (a) fine-tuning or (b) prompting - can encourage models to attend more to abstract meaning, thereby improving generalization and overall performance. As illustrated in Figure 1, we add AMRs to either the training data or prompts then evaluate how the addition of AMR affects generalization performance. We find that AMRs generally hinder performance in both fine-tuning and prompting settings, with the exception of prompting on HANS. However, this improvement appears to stem from AMRs amplifying surface-level differences rather than capturing deeper semantic meaning.

2 Related Work

NLI (Dagan et al., 2010) is a hallmark task demonstrating model's ability to understand natural language. Select neural models like `BERT` (Devlin et al., 2019) and `RoBERTa` (Liu et al., 2019) trained on datasets like Multi-genre NLI (MNLI; Williams et al., 2018) and Stanford NLI (SNLI; Bowman et al., 2015) provide test-set performance close to that of humans (Nangia and Bowman, 2019), but the near-human performance on MNLI has been attributed to models optimizing on the spurious correlations between lexical items and labels in the data (Poliak et al., 2018; McCoy et al., 2019; Gururangan et al., 2018; Serrano et al., 2023). The same models that excel in test-set performance suffer from poor generalization to other datasets that represent the same task (Zhou et al., 2020; McCoy et al., 2020; Delbari and Pilehvar, 2025).

Several prior approaches have incorporated logical representations into NLI, for example by combining neural encoders with logical reasoning modules (Chen et al., 2021), training natural logic theorem provers (Abzianidze, 2020), extracting phrase correspondences via natural deduction proofs (Yanaka et al., 2018), or constraining large language models with natural logic inference patterns (Noble et al., 2025). While these works rely on task-specific inference rules or specialized proof systems, our use of AMRs differs in that AMRs provide a broad, task-agnostic semantic abstraction without requiring dedicated engineering. Once an AMR parser is available, AMRs can be used as direct inputs to pretrained models such as BERT (Section 3.2 and ChatGPT (Section 3.3), enabling structured input with minimal task-specific engineering.

LLMs and in-context learning have been used to tackle NLI and generalization in it, with mixed results; Webson and Pavlick (2022) show that the content of prompts do not significantly influence LLMs' performance in NLI tasks, while Kavumba et al. (2023); He et al. (2024) use chain-of-thought reasoning and natural language explanations to improve NLI performance and generalization. However, Zhong et al. (2023) report that its NLI performance is still only comparable to much smaller encoder-only models like `BERT` and `RoBERTa` (Devlin et al., 2019; Liu et al., 2019), leaving adversarial NLI an ongoing area of research.

Recent work on AMRs has set out to utilize AMR graphs for a variety of downstream tasks, including summarization and information extraction (see Wein and Opitz (2024); Sadeddine et al. (2024) for comprehensive overviews). AMRs excel in capturing structure-dependent meaning (Leung et al., 2022) and have shown particular promise in meaning-sensitive tasks such as debiasing translationese (Wein and Schneider, 2024), style transfer (Hua et al., 2023), and sentence-level manipulation (Li et al., 2025), especially when used in conjunction with fine-tuned models.

To the best of our knowledge, Opitz et al. (2023) represents the only prior effort to incorporate AMRs into NLI, and do so for the purpose of interpretable NLI evaluation. They find that metrics based on AMR are robust unsupervised representations of premise-hypothesis relationships when used alongside neural representations like `BERT`.

3 Data & Experiments

3.1 Data & Models

In these experiments, we use two datasets: MNLI (Williams et al., 2018) and HANS (McCoy et al., 2019). MNLI is a crowdsourced dataset, with a test set that is not available to the public. We follow prior work (Wang et al., 2018; Devlin et al., 2019) in taking one of its two developmental splits as the evaluation dataset. Specifically, we take the `matched` developmental set to use as our evaluation dataset. The training dataset includes 297k sentence pairs, while the evaluation set contains around 10k pairs. HANS is a template-based evaluation dataset, with 30k examples. Unlike MNLI and other NLI datasets, its label space consists of only two labels–`entailment` and `non-entailment`. We follow prior work (McCoy et al., 2020; Min et al., 2020) in collapsing the model's `neutral` and `contradiction` predictions to the single `non-entailment` label when calculating evaluation metrics, to accommodate the two-class label space of HANS.

We use an off-the-shelf AMR parser from `amrlib` [2] to parse all the sentences from the two datasets we use. The model is `BART-large` (Lewis et al., 2019) fine-tuned on AMR 3.0 (Knight et al., 2021). While parsers with higher reported scores exist (e.g. Bevilacqua et al., 2021), we follow Uhrig et al. (2021); Opitz et al. (2023) in selecting an `amrlib` parser for ease of implementation.

We manually perform a small sanity check over a subset of generated AMRs to verify that AMR parses are acceptable, but do not perform a comprehensive quality check over the entire dataset. We observe that the AMRs produced for sentences in the HANS dataset are generally acceptable, likely benefiting from the sentences' simple structure and short length, though certainly the generated AMRs contain noise; the sentences in MNLI are longer and more complex.

3.2 Experiment 1: Can fine-tuned models benefit from AMR in NLI?

We train three sets of `BERT-base` models, augmented with AMR information to perform our experiment. We incorporate AMR in three ways: (1) linearized AMR is concatenated to text input (+AMR as text); (2) graph neural network representation of AMR is concatenated to text representation (+AMR as graph); and (3) just the linearized AMR is used as text input (AMR as text only).

We adopt the setup and hyperparameters of previous work in MNLI fine-tuning and HANS evaluation (McCoy et al., 2020; Min et al., 2020). We take the `bert-base-uncased` model and fine-tune for 3 epochs with a learning rate of 2e-5. While we opt to follow prior work, we note that longer fine-tuning beyond 10 epochs at the same learning rate significantly improves HANS performance in all settings. Each label prediction is made from the `[CLS]` token's final layer embedding. While the setup is equivalent to those from prior work, we implement the setup in a more modern, current stack. Due to updates in the hardware and software since prior work, slight changes in the resulting model weights are possible. To control for such an effect, we perform a sanity check via baseline in-distribution test set evaluation. Finally, we integrate AMR into the models as text, via *linearization*, removing all newlines and whitespace sequences longer than length two.

3.3 Experiment 2: Can prompt-based models benefit from AMR in NLI?

In this experiment, we evaluate whether incorporating AMRs improves LLMs' performance on NLI, on both the MNLI and HANS dataset, the latter of which remains challenging even after fine-tuning. Jin et al. (2024) find that only instruction-tuned GPT models are capable of reliably processing AMRs. We therefore restrict our evaluation to `GPT-4o` (Hurst et al., 2024) in zero-shot and 5-shot settings.

We use the following prompt template:

> *You are a helpful assistant trained to determine whether a hypothesis logically follows from a premise. Respond with 'Yes' or 'No'.*
> *Premise: [X].*
> *Hypothesis: [Y].*

Where [X] and [Y] are replaced with the premise and hypothesis in question. The prompt applies to both zero and few-shot settings. We incorporate no additional details or explanations about the task (NLI), the datasets (MNLI and HANS), or the AMRs in our prompt, to best measure the LLM's ability to use representations of meaning for NLI, rather than perform in-context learning. However, it is possible that the model perform better with additional context on the task, dataset, or AMRs.

We test three input conditions: (a) sentence only; (b) AMR only; and (c) sentence + AMR. Label preprocessing follows the same procedure as in the fine-tuning setup for MNLI. In the 5-shot setting, we randomly sampled 5 examples from the training set of each data set. We set the temperature to 0 to ensure deterministic outputs.

4 Results & Discussion

4.1 Experiment 1

We report the accuracies of our fine-tuning models with and without AMRs in Table 1. We report numbers from prior work (McCoy et al., 2020;

---

[2] `https://github.com/bjascob/amrlib-models/releases/tag/parse_xfm_bart_large-v0_1_0`

Model	**MNLI**	**HANS**
Chance	0.33	0.50
Baseline (McCoy et al., 2020)	0.84	0.57
+Syntactic aug (Min et al., 2020)	0.84	0.65
Ours		
Baseline reproduction (text only)	0.84	0.52
+AMR as text	0.83	0.47
+AMR as graph	0.84	0.49
AMR as text only	0.74	0.51

Table 1: Performance comparison with and without AMR on HANS and MNLI test sets in the fine-tuning setting. Both datasets measure accuracy.

Model	MNLI	HANS
Chance	0.33	0.50
ChatGPT-3.5		
Zhong et al. (2023)	0.89	-
He et al. (2024)	-	0.75
Ours (ChatGPT-4o)		
Text only	**0.91**	0.82
+AMR	0.75	**0.87**
AMR only	0.68	0.70

Table 2: Performance comparison with and without AMR on HANS and MNLI test sets in the LLM zero-shot prompting setting.

Min et al., 2020) in addition to our experiments to serve as comparison baselines and to ensure our setup is correct. Our reported numbers are an average across 10 runs with varying seed.

As shown in Table 1, AMR augmentation does not yield improvements in MNLI performance, nor HANS generalization. Perhaps analogously to previous data-driven attempts at improving generalization (Clark et al., 2019; Min et al., 2020; Yaghoobzadeh et al., 2021), additional AMR information as either text or graph does not affect MNLI performance. Analysis of their confusion matrices reveals AMR adds or subtracts little in terms of MNLI label decision boundary. On HANS performance, We discuss two main findings.

Standalone AMR input for classification intensifies heuristics favoring the entailment label. AMR-only models predict the entailment label for 98.3% of HANS examples, compared to the baseline models at 94.7%. We attribute this to an intensified version of the baseline models' heuristic correlating overlap between the hypothesis and premise to the entailment label, dubbed the lexical overlap heuristic (McCoy et al., 2019). We note this is concurrent with a still competitive MNLI performance, at 84%. We discuss this phenomenon in more detail in Sections A.1 and A.2.

Mixing AMRs and text leads to more (false) negative predictions in novel data. On the other hand, combining AMR information with text strongly affects HANS label decision boundaries in the opposite direction, overriding various shallow heuristics that favor the entailment label observed in McCoy et al. (2020) and in our baseline and AMR-only experiments. Our `+AMR as text` models 86.6% of HANS examples, and `+AMR as graph` models 86.9%, even predicting non-entailment on highly overlapping examples. We attempt to disentangle the effects of AMRs and text in a combined representation in Section A.2, where we find that while AMR can be used to perform NLI, it is less effective than text input and combining the two introduces new artifacts that are more difficult to interpret.

4.2 Experiment 2

The results for prompting with `GPT-4o` are shown in Table 2. We report only the zero-shot results in the main text, as they yield similar overall performance and prediction patterns. Results for the five-shot setting are provided in Section D. Two main observations emerge.

AMRs increase (false) negative predictions. As shown in the table, model performance is consistently lowest when prompted with AMRs alone, while including the original sentence improves results. We find this is because AMRs lead models to make more negative predictions (see Section B). To test this statistically, we fit a logistic regression model predicting non-entailment using SMATCH++ (Opitz, 2023) between hypothesis and premise AMRs and data source (gold vs. predicted). A significant negative interaction (β = -0.042, p < 2e-16) shows that SMATCH similarity influences model predictions more than gold labels.

Further analysis reveals that AMR's sensitivity to surface-level lexical and syntactic variation leads to low structural overlap between semantically equivalent expressions,[3] misleading the model toward non-entailment. This also explains why, on the HANS test set, prompts that include both the sentence and its AMR lead to the highest rate of negative predictions: the AMR repre-

---

[3] See Section C for an example.

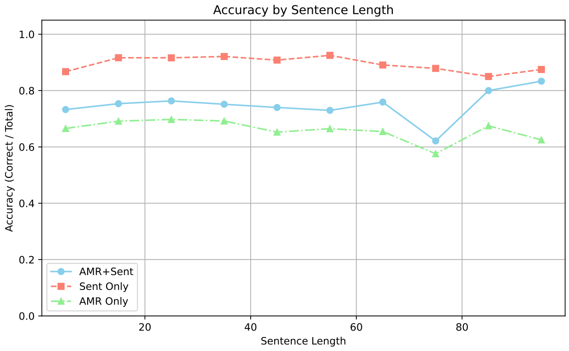


Figure 2: Accuracy of three prompt settings across different sentence lengths on MNLI.

sentation amplifies subtle differences between two otherwise similar strings, making semantic mismatches more salient and pushing the model toward rejecting entailment. Such nuanced contrasts are what HANS is designed to probe in language models, prompting `GPT-4o` to overpredict non-entailment.

AMR does not lead to more robust performance with longer sequence length. Opitz et al. (2023) reported that incorporating AMRs improves robustness in NLI prediction. We investigate whether this finding holds for LLMs. Specifically, we plot accuracy across NLI examples binned by total sequence length (premise + hypothesis). For sequences exceeding 100 words, we group them into a single bin due to their sparsity.

As shown in Figure 2, when `GPT-4o` is prompted with both sentence and AMR inputs, accuracy slightly increases for inputs longer than 80 words. However, this performance remains lower than that of sentence-only prompts across most length bins. We find no evidence that AMR-only prompts enhance robustness to longer sequences.

4.3 Summary

Our fine-tuning experiments suggest that AMR-only models are still susceptible to heuristics. We also observe that combining text with AMR as both graph and text is challenging and results in a strong preference towards the non-entailment label, even for highly overlapping, entailing examples.

Our LLM experiments showcase similar preference towards the non-entailment label. This suggests that AMRs effectively highlight subtle distinctions between minimal pairs, explaining improved **HANS** performance. However, for simpler examples, this heightened contrast can cause the model to overpredict *No*, even for entailing sentence pairs.

5 Conclusion

In this work, we investigate whether AMRs can help PLMs on the task of natural language inference. Specifically, across both fine-tuning and prompting settings, we evaluate whether incorporating AMRs improves entailment classification.

We find that our implementations of AMR integration does not improve performance in fine-tuning, and only lead to slight gains in zero-shot prompting with `GPT-4o`. Importantly, ablation analyses reveal that these gains are not due to deeper semantic understanding, but rather to AMRs exaggerating surface-level differences, which in some cases mislead the model to predict *non-entailment* where entailment holds. Overall, our results suggest that while AMRs offer a promising abstraction mechanism, their integration with LLMs requires careful design to avoid reinforcing shallow heuristics rather than promoting robust reasoning.

Limitations and Future Work

This study focuses on two datasets (MNLI and HANS) and explores a limited set of prompting and fine-tuning configurations. For fine-tuning, we adopt a single AMR linearization strategy; in the prompting setting, we test one prompt template with different conditions. While alternative prompts for zero-shot inference may yield better performance (e.g., Kavumba et al., 2023), our consistent experimental setup enables fair comparisons across conditions. Nonetheless, the findings may not generalize to other inference tasks, domains, or prompting strategies.

Future work could explore more diverse linearization formats, prompt designs, and integration strategies that align AMR structure more directly with model attention or reasoning processes.

Encoder-based models have been shown to be sensitive to minor perturbations in input (Sinha et al., 2021; Jin et al., 2020), and prior work integrating AMR graphs into neural models have used a variety of formats (Wein and Opitz, 2024). Thus, in addition to Python-like and natural language-like representation of AMR's structure (Srivastava et al., 2025; Srivastava and Yao, 2025; Dutt et al., 2025), carefully designing how hierarchical devices in AMRs (e.g. variable names, parentheses, indents, and newlines) could be represented in the embedding space of encoder-only models may be worth further investigations.

Finally, investigating how AMRs interact with LLM decoding beyond surface augmentation may help unlock their full potential in meaning-sensitive tasks.

Responsible Research Statement

We use `ChatGPT-4o` (Hurst et al., 2024) as a coding assistant during the implementation of our experiments, in addition to as a natural language processor.

Acknowledgments

This work was developed on a course project for Advanced Semantic Representation at Georgetown University. We thank Nathan Schneider and our classmates from the course for their helpful comments. A portion of experiments was conducted on the `gcp-gu-hpc-cli` cluster within the High Performance Computing at Georgetown University infrastructure.

References

Lasha Abzianidze. 2020. Learning as abduction: Trainable natural logic theorem prover for natural language inference. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 20–31, Barcelona, Spain (Online). Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Zeming Chen, Qiyue Gao, and Lawrence S. Moss. 2021. NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88, Online. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches – erratum. *Natural Language Engineering*, 16(1):105–105.

Zahra Delbari and Mohammad Taher Pilehvar. 2025. Beyond accuracy: Revisiting out-of-distribution generalization in NLI models. In *The SIGNLL Conference on Computational Natural Language Learning*.

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

Ritam Dutt, Shounak Sural, and Carolyn Rose. 2025. Can dependency parses facilitate generalization in language models? a case study of cross-lingual relation extraction. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 338–358, Albuquerque, New Mexic, USA. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2024. Using natural language explanations to improve robustness of in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13477–13499, Bangkok, Thailand. Association for Computational Linguistics.

Yilun Hua, Zhaoyuan Deng, and Kathleen McKeown. 2023. Improving long dialogue summarization with semantic graph representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13851–13883, Toronto, Canada. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Zhijing Jin, Yuen Chen, Fernando Gonzalez Adauto, Jiarui Liu, Jiayi Zhang, Julian Michael, Bernhard Schölkopf, and Mona Diab. 2024. Analyzing the role of semantic representations in the era of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3781–3798, Mexico City, Mexico. Association for Computational Linguistics.

Pride Kavumba, Ana Brassard, Benjamin Heinzerling, and Kentaro Inui. 2023. Prompting for explanations improves adversarial NLI. is this true? {yes} it is {true} because {it weakens superficial cues}. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2165–2180, Dubrovnik, Croatia. Association for Computational Linguistics.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and 1 others. 2021. Abstract meaning representation (amr) annotation release 3.0.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Hongji Li, Andrianos Michail, Reto Gubelmann, Simon Clematide, and Juri Opitz. 2025. Sentence smith: Formally controllable text transformation and its application to evaluation of text embedding models. *arXiv preprint arXiv:2502.14734*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy. Association for Computational Linguistics.

Bill Noble, Rasmus Blanck, and Gijs Wijnholds. 2025. In the mood for inference: Logic-based natural language inference with large language models. In *Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA)*, pages 33–47, Bochum, Germany. Association for Computational Linguistics.

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Juri Opitz, Shira Wein, Julius Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 275–283, Nancy, France. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A survey of meaning representations – from theory to practical utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2877–2892, Mexico City, Mexico. Association for Computational Linguistics.

Sofia Serrano, Jesse Dodge, and Noah A. Smith. 2023. Stubborn lexical bias in data and models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8131–8146, Toronto, Canada. Association for Computational Linguistics.

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. 2021. Perturbing inputs for fragile interpretations in deep natural language processing. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 420–434, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. Instruction-tuning llms for event extraction with annotation guidelines. *arXiv preprint arXiv:2502.16377*.

Saurabh Srivastava and Ziyu Yao. 2025. Revisiting prompt optimization with large reasoning models-a case study on event extraction. *arXiv preprint arXiv:2504.07357*.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

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

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Lost in translationese? reducing translation effect using Abstract Meaning Representation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–765, St. Julian's, Malta. Association for Computational Linguistics.

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

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki. 2018. Acquisition of phrase correspondences using natural deduction proofs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 756–766, New Orleans, Louisiana. Association for Computational Linguistics.

Peiran Yao, Kostyantyn Guzhva, and Denilson Barbosa. 2024. Semantic graphs for syntactic simplification: A revisit from the age of LLM. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 105–115, Bangkok, Thailand. Association for Computational Linguistics.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

A Fine-tuning Error Analyses

A.1 Intensified subsequence overlap heuristic with AMR

Compared to text-only MNLI models which are known to incorrectly correlate lexical and sequence overlap to the entailment label (McCoy et al., 2019, 2020; Min et al., 2020), the AMR-only models favor the entailment label even more. The model's preference toward the entailment label results in the AMR-only models consistently predicting non-entailment for around 98% of HANS examples.

It is less likely that the model adopts the subsequence and constituency heuristic, as no text subsequences or constituencies are provided in the training dataset–only AMR parses are provided as input. However, it is possible that a new heuristic had formed. Consider the following two versions of premise-hypothesis pairs:

- **Premise:** *The judge and the president advised the scientist.*

```
(a / advise-01
     :ARG0 (a2 / and
          :op1 (p / person
               :ARG0-of (h / have-org-role
                   -91
                    :ARG3 (j / judge-01)))
          :op2 (p2 / person
               :ARG0-of (h2 / have-org-
                   role-91
                    :ARG2 (p3 / president)
                        )))
     :ARG1 (s / scientist))
```

- **Hypothesis 1, label=non-entailment:** *The scientist advised the judge.*

```
(a / advise-01
     :ARG0 (s / scientist))
     :ARG1 (p / person
          :ARG0-of (h / have-org-role-91
               :ARG3 (j / judge-01)))
```

- **Hypothesis 2, label=entailment:** *The judge advised the scientist.*

```
(a / advise-01
     :ARG0 (p / person
          :ARG0-of (h / have-org-role-91
               :ARG3 (j / judge-01)))
     :ARG1 (s / scientist))
```

The premise and hypothesis AMRs exhibit significant overlap, namely in variables `p`, `h`, `j`, `s`. Given sufficiently many pairs similar to `Premise-Hypothesis 2` in the training set, the model may optimize to correlate variable overlap to the entailment label. Then, when the model predicts on `Premise-Hypothesis 1` pair, instead of considering the semantic structure, it may attend to the significant variable overlap, and predict `entailment`, which is the incorrect answer.

Train setting	Text eval	+AMR eval	AMR only eval
Text	0.84	0.47	
+AMR	0.53	0.83	0.36
AMR only		0.44	0.74

Table 3: MNLI accuracy of our trained models evaluated on each setting. Chance performance is 0.33.

Train setting	Text eval	+AMR eval	AMR only eval
Text	0.96	0.97	
+AMR	0.25	0.13	0.85
AMR only		0.99	0.98

Table 4: Percentage of HANS examples where our trained models evaluated on each setting predict entailment.

A.2 Cross-setting evaluation analysis

To disentangle the effects of text and AMR in `+AMR` models, we evaluate models in not only their own evaluation setting, but in other settings as well. We do not evaluate all models on all settings. Instead, we measure performance on reasonable train-evaluation setting pairs–we do not evaluate `text only` models on `AMR only` settings, and vice versa. `+AMR` models undergo all evaluation settings; all models undergo evaluation in the `+AMR` setting. In this cross-setting evaluation scheme, we do not consider the `+AMR as graph` setting.

First, we observe that cross-evaluation models still perform above chance in MNLI evaluation (0.33), as seen in Table 3, which indicates both text and AMR knowledge can be leveraged despite noise from unseen form. The `+AMR` models' single-mode (text only or AMR only) MNLI accuracies, together with the lower performance of `AMR-only` models compared to `text-only` models indicate that AMR information is more difficult to acquire and use than text input.

Second, we observe that bias towards entailment or non-entailment in MNLI and HANS is strongly correlated, given train-evaluation mismatch (m = 0.6, R^2 = 0.87). Cross-setting evaluation results support the case of a newly developed heuristic for `AMR only` models, as single-

Prompt	MNLI	HANS
Text + AMR	+2,274	+1,759
AMR only	+2,848	+1,393

Table 5: Increase in the number of negative predictions compared to the sentence-only prompt condition.

mode models overwhelmingly predict entailment in HANS examples even when evaluated on `+AMR` settings, both at above 96%, as seen in Table 4.

On the other hand, it is difficult to pinpoint the cause of the tendency to predict non-entailment in dual-mode models predicting on input containing text. We observe that dual-mode models predict non-entailment for entailing adverbial sentences whose AMRs highly overlap, as shown below:

- **Premise:** *Clearly the bankers waited.*

```
(w / wait-01
    :ARG1 (b / banker)
    :ARG1-of (c / clear-06))
```

- **Hypothesis, label=entailment, pred=non-entailment:** *The bankers waited.*

```
(w / wait-01
    :ARG1 (b / banker)
```

B LLM prediction statistics

The results are reported in Table 5.

C LLM setting error analysis: Example

For example, while the premise *everything you're looking for is available* is semantically equivalent to the hypothesis *everything can be found*, the AMRs for these sentences differ substantially due to lexical choices (e.g., *look for* vs. *find*) and syntactic voice (active vs. passive). The resulting SMATCH++ F-score (Opitz, 2023) between the two graphs is only 27.7.

- **Premise:** *Enter the realm of shopping malls, where everything you're looking for is available without moving your car.*

```
(e / enter-01
      :ARG0 (y / you)
      :ARG1 (r / realm
            :mod (m / mall
                  :mod (s / shop-01)
                  :location-of (a / available
                       -02
                        :ARG2 (e2 / everything
                              :ARG1-of (l /
                                   look-01
                                    :ARG0 y))
                  :manner (m2 / move-01
                        :polarity -
                        :ARG0 y
                        :ARG1 (c / car
                              :poss y))))
                                 ))
```

- **Hypothesis:** *Everything can be found inside a shopping mall.*

```
(p / possible-01
      :ARG1 (f / find-01
            :ARG1 (e / everything)
            :location (ii / inside
                  :op1 (m / mall
                        :purpose (s / shop-01)
                           ))))
```

D 5-Shot Prompting Result

We report the results of our 5-shot prompting experiments in Table 6.

Prompt	MNLI	HANS
Text only	0.89	0.82
+AMR	0.75	0.88
AMR only	0.69	0.67

Table 6: Performance comparison with and without AMR on HANS and MNLI test sets in the LLM five-shot prompting setting.

Explanations *explained*. Influence of Free-text Explanations on LLMs and the Role of Implicit Knowledge

Andrea Zaninello[1,2], Roberto Dessì[3], Malvina Nissim[4], Bernardo Magnini[2]
[1]Free University of Bolzano, Italy
[2]Fondazione Bruno Kessler, Italy
[3]Not Diamond, San Francisco, USA
[4]University of Groningen, Netherlands
`azaninello@fbk.eu`, `m.nissim@rug.nl`, `magnini@fbk.eu`

Abstract

In this work, we investigate the relationship between the quality of explanations produced by different models and the amount of implicit knowledge they are able to provide beyond the input. We approximate explanation quality through accuracy on a downstream task with a standardized pipeline (GEISER) and study its correlation with three different association measures, each capturing different aspects of implicitness, defined as a combination of relevance and novelty. We conduct experiments with three SOTA LLMs on four tasks involving implicit knowledge, with explanations either confirming or contradicting the correct label. Our results demonstrate that providing quality explanations consistently improves the accuracy of LLM predictions, even when the models are not explicitly trained to take explanations as input, and underline the correlation between implicit content delivered by the explanation and its effectiveness.[1]

1 Introduction

Large Language Models (LLMs) excel at numerous language processing tasks, including text generation, translation, and question answering (Touvron et al., 2023; OpenAI, 2023). Still, understanding their reasoning is challenging, hindering trust and adoption in high-stakes domains (Hase et al., 2020; Kaneko and Okazaki, 2023; Kotonya and Toni, 2020; Atanasova et al., 2020). One approach towards "intrinsic explainability" is to have LLMs generate explanations for their predictions. Existing methods, like pipeline models (Wiegreffe et al., 2020) and self-rationalizing models (Lei et al., 2016), often focus on extractive rationales suitable for information extraction (Jacovi et al., 2021). However, complex reasoning tasks require free-text explanations, especially when implicit knowledge is involved (Wiegreffe et al., 2021). Also, generating explanations raises concerns about their faithfulness, as LLMs might produce plausible-sounding explanations with no genuine connection to their reasoning (Narang et al., 2020). This is particularly problematic for implicit knowledge, which relies on the model's internal representations of the world (McClelland et al., 2020).

With the rise of retrieval-augmented generation (RAG, Lewis et al. (2020)), language models are increasingly supplemented with external information, such as explanations, retrieved from knowledge bases or provided via in-context learning (ICL). The effectiveness of these approaches depends on the quality of the retrieved or injected text, which serves as additional context for the model's reasoning. While traditional RAG studies focus on improving retrieval mechanisms (e.g., optimizing factual correctness), less attention has been paid to evaluating the quality of explanations used in these frameworks. Recent work by He et al. (2024) shows that augmenting ICL with natural language explanations (NLEs) improves model robustness. However, their study focuses on performance benefits rather than the quality of different explanation types, and their evaluation is limited to downstream accuracy without assessing what makes an explanation effective in guiding a model's decision.

Our work addresses this gap by providing a principled evaluation of explanation quality, particularly in sentence pair reasoning tasks, measured by downstream task performance. Moreover, we show that explanation effectiveness correlates with the degree of *implicit content*, i.e., novel yet relevant information they provide. We test this hypothesis by examining the relationship between explanation effectiveness and three metrics approximating novelty and relevance, and show that they have high, yet different correlation with explanation quality according to the examined task.

The main contributions of this paper are:

---

[1] Code and data available here `github.com/andreazaninello/geiser`.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 212–224
November 8-9, 2025 ©2025 Association for Computational Linguistics

- we propose GEISER, a standardized pipeline to evaluate the effectiveness of different types of explanations using LLM relation predictions on tasks involving varying degrees of implicit reasoning and external knowledge;

- using the proposed pipeline, we report experimental results on different kinds of explanations (human- and machine-generated), across three LLMs, four tasks and two languages;

- through our analysis, we introduce "implicit knowledge" as a key factor of explanation quality, and study different metrics to estimate it, showing its correlation with explanation effectiveness.

2 Related Work

The role of explanations in NLP has been extensively studied. For instance, Cambria et al. (2023) provide a comprehensive survey of natural language **explanation generation** approaches, and Hartmann and Sonntag (2022) examine the benefits of explanations for improving NLP models. Paranjape et al. (2021) focus on template-based explanations, while Lampinen et al. (2022) and Ye and Durrett (2022) highlight the advantages of in-context explanations for complex reasoning tasks. Jansen et al. (2016) provide a comprehensive characterization of different kinds of explanations, each one with different insight into model behavior.

Traditionally, **explanation quality** has been assessed using automated metrics like BLEU (Papineni et al., 2002), ROUGE (ROUGE, 2004), or BERT-Score (Zhang et al., 2019), which compare outputs to human-written references. However, these metrics may not fully capture explanation quality or align with human judgment, and collecting human references is often costly. More recently, (human) *simulatability scores* have emerged as an alternative to overlap metrics, based on the idea that explanation quality can be defined as the "utility to an end-user" (Kim et al., 2016). This approach evaluates how explanations improve predictive performance on downstream tasks rather than overlap with ground truth explanations and, while humans were initially the predictors (Wiegreffe et al., 2021), trained models now automate this process, showing strong correlations with human judgments (Hase et al., 2020). For example, Pruthi et al. (2022) measures explanation quality by training a student model on teacher-generated explanations for downstream tasks.

Prior work has largely focused on eliciting explanations from models or evaluating them based on task performance, our work shifts the focus toward understanding how explanations can reveal implicit knowledge, offering a novel perspective on explanation quality assessment. While, to the best of our knowledge, there are no previous works addressing **implicit content measures** directly, in the context of information retrieval, *relevance* and *novelty* have been recognized as key aspects of *novelty detection tasks* (Ghosal et al., 2022, 2018), and similarly to us exploit Textual Entailment (Bentivogli et al., 2011) for sentence level novelty mining. Metrics such as the cosine similarity between high-dimensional embeddings has been traditionally used to quantify semantic similarity of texts, but has also been recently questioned as a faithful representation (Steck et al., 2024). Other works, on the other hand, have focused on estimating the *causal strength* between textual fragments, and proposed learned metrics such as CEQ (Du et al., 2022) or CESAR (Cui et al., 2024), in the attempt to improve more simplistic yet effective metrics such as Pointwise Mutual Information (PMI).

3 Methodology

We address the problem of explaining the semantic relationship between two textual fragments under the assumption that the relationship involves implicit knowledge, and the hypothesis that explanations eliciting more implicit knowledge represent higher-quality explanations.

3.1 Explanatory task

Given a pair of sentences $< s_1, s_2 >$, and a semantic relation r between s_1 and s_2 (e.g., s_1 temporally precedes s_2, s_1 is caused by s_2, s_1 contradicts s_2, etc.). The task consists in a model M_1 generating an explanation e_i for the relation r and then in a model M_2 using the explanation e_i to predict the relation r for the same sentence pair, when r is not given. The goal is to support the hypothesis that using explanations results in better predictions, and that an increase in prediction accuracy corresponds to higher explanation effectiveness, as well as investigate the correlation between explanation quality, implicit information elicitation, and relation prediction.

3.2 The GEISER Pipeline

To estimate the quality of the explanations, we propose GEISER (Generation and evaluation of Explanations for Implicit SEmantic Relations) a three-step methodology inspired by work on human simulatability scores.

Step 1: Generate Explanations with M1. Given an explanatory task, we ask a model M_1 to generate a set of possible explanations E for the semantic relation r_c for the sentence pair $< s_1, s_2 >$. We assume ground truth relations R_c from human annotators, as they guarantee explanations consistent with the actual semantic relations of the sentence pair.

$$M_1(s_1, s_2, r_c) \Rightarrow E$$

As we are interested in comparing different explanations $E = \{e_1, e_2, \dots e_n\}$ for the same sentence pair and the same relation r_c (e.g., a counterfactual explanation vs. a why-explanation) each explanation e_i is generated independently, prompting a generative model for each specific explanation type. In Section 5 we define in detail the set E of explanation types.[2]

Step 2: Predict Relation with M2. Here, model M_2 is asked to predict a semantic relation r_p between s_1 and s_2 given one individual explanation e_i in E, injected into the input along with the sentence pair. Adding one explanation e_i is meant to potentially add new information, implicit in s_1 and s_2, that can help the model M_2 predict the correct relation r_c.

$$M_2(s_1, s_2, e_i) \Rightarrow r_p$$

The two models used in step 1 and step 2, M_1 and M_2, might be the same model, in which case the goal is to assess the self-consistency of the model (generate the explanation and then use it for prediction), or two different models, in which case the goal is to have an independent assessment of the explanation quality. M_1 must be a generative model, as it has to produce the set of explanations E, while M_2 is a generative model performing a classification task.

Step 3: Evaluate M1's Explanations through M2's performance. Our final goal is to assess the quality of the explanations in E generated by M_1. Intuitively, the quality of an explanation e_i depends on its ability to provide useful content to solve a relation prediction task: the more e_i is useful to the model M_2 to predict the correct relation r_c, the better its *effectiveness*, taken as a proxy of the quality of e_i. Accordingly, here we assume that the M_2 performance is an indicator of the explanation effectiveness, such that better explanations are those that contribute to better prediction accuracy. Given an explanation e_i in the set E, its effectiveness relative to a model M_2 is given by the ability of the model to predict a relation r_p that approximates the correct relation r_c for a given sentence pair.

$$\textit{Effectiveness}(e_i, M_2) = r_p \approx r_c$$

In practice, overall accuracy of a model M_2 on a relation prediction task is used as a proxy metric for explanation *effectiveness*. There are two interesting aspects to be considered. First, the difference between the relation prediction of the M_2 model without and with e_i: this is an indicator of the absolute effectiveness of a certain explanation. Second, as an aggregation metric, the relative ranking of all explanations in $E_t \in E$ given by the M_2 accuracy according to their type and how they were generated: this will give us an indication of whether an explanation type or a generative model is better (i.e., more effective) than another.

3.3 Measuring Implicit Content via Explanation–Input Association Measures

We want to explore whether better explanations are those that are able to introduce highly relevant implicit knowledge, i.e., not present in the sentence pair $< s_1, s_2 >$, that the M_2 model can use for predicting r_p. Intuitively, a good explanation for an implicit knowledge-based relationship should maximize both its *novelty*, i.e., it has to bring new, implicit content with respect to $< s_1, s_2 >$, and its *relevance* with respect to $< s_1, s_2 >$, i.e., it has to be grounded to entities and events mentioned in the sentences (Ghosal et al., 2018).

As a first step towards validating this hypothesis, we define the amount of implicitness of an explanation e_i as the combination of *relevance* and *novelty* of e_i with respect to a sentence pair $< s_1, s_2 >$.

We operationalise the implicit content calculations comparing three different association measures between the input sentences $< s_1, s_2 >$ and the explanation e_i: *Causal Strength (CS)*, *Entailment Probability*, and *Cosine Similarity*. These are

---

[2] To keep under control our experimental setting, we assume only one semantic relation r_c for a given sentence pair.

intended to variously reflect how well an explanation relates to the input sentences s_1 and s_2, while bringing new, potentially useful information.

Causal Strength

Firstly, we consider the metric of CAUSAL STRENGTH (CS) as proposed by Cui et al. (2024), which—in its original formulation—aggregates token-level associations between a *cause* sequence C (length n) and an *effect* sequence E (length m). However, following Du et al. (2022), in our setting the cause sequence C is obtained by taking the maximum value obtained by calculating the causal strength either between a. the concatenation of s_1 and the explanation and the effect sequence E (corresponding to s_2) or 2. s_1 as C and the concatenation of the explanation and s_2 as E. The method uses causal token embeddings from a BERT model pre-trained on a cause–effect corpus, and attention weights to focus on the most relevant token pairs.

Formally, the score is defined as:

$$\mathrm{CS}(C, E) = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \left| \frac{\mathbf{c}_i^\top \mathbf{e}_j}{\|\mathbf{c}_i\| \|\mathbf{e}_j\|} \right|$$

where $\mathbf{c}_i$ and $\mathbf{e}_j$ are the token embeddings of C and E, and a_{ij} are normalized attention weights over token pairs ($\sum_{i,j} a_{ij} = 1$). This attention-weighted absolute cosine similarity captures both semantic alignment and token-level causal informativeness.

Intuitively, this metric emphasizes token pairs that are both semantically aligned (via cosine similarity) and deemed important by the attention mechanism—highlighting explanatory tokens that are both novel and causally relevant. This implementation, known as CESAR (Cui et al., 2024), yields more robust predictions of causal strength changes when additional information (e.g., supporters or defeaters) is introduced.

Entailment Probability

Secondly, we consider the probability of entailment, also referred to as NATURAL LANGUAGE INFERENCE (NLI), calculated via a pre-trained NLI model. Given an explanation exp, an input statement s_1, and a target statement s_2, we define:

$$\mathrm{NLI}(\mathrm{exp}, s_1, s_2) = P_{\mathrm{NLI}}\big(s_1 \wedge \mathrm{exp} \models s_2\big).$$

This directional measure, while theoretically capturing both relevance and novelty, may in practice favour relatedness over new information, and its reliability is limited by the accuracy of the underlying NLI model.

Cosine Similarity

Thirdly, we consider simply comparing the COSINE SIMILARITY (COS) of the embedding vectors $\mathbf{e}_{\mathrm{exp}}$ and $\mathbf{e}_{s_1,s_2}$:

$$\mathrm{COS}(\mathbf{e}_{\mathrm{exp}}, \mathbf{e}_{s_1,s_2}) = \frac{\mathbf{e}_{\mathrm{exp}} \cdot \mathbf{e}_{s_1,s_2}}{\|\mathbf{e}_{\mathrm{exp}}\| \, \|\mathbf{e}_{s_1,s_2}\|}.$$

This measure captures semantic relatedness but not novelty, and can be sensitive to embedding behaviors (e.g., scale-invariance may obscure frequency effects).

In summary (Table 1), CS offers an interpretable balance of novelty and relevance; NLI aligns closely with the conceptual role of explanations, though reliability is tied to model strength and may not fully reflect novelty; COS, on the other hand, is easy to compute but lacks novelty sensitivity. For this reason, we hypothesize that CS should better align with explanation effectiveness, as defined above, and thus positively correlate with accuracies at the system level.

Measure	**Relevance**	**Novelty**	**Reliability**
CS	yes	yes	Corpus-based, robust
NLI	yes	no	Theoretical, model-limited
COS	yes	no	Fast, but surface-level

Table 1: Overview of implicit content measures and their features.

4 Tasks and Datasets

We use four datasets that propose tasks involving different kinds of reasoning and eliciting implicit or external knowledge to various extents. All datasets provide either human-generated or human-collected and curated explanations (which we use as the `gold` baseline, see Section 5)[3]

e-SNLI (Natural Language Inference). A version of the Stanford Natural Language Inference (SNLI) corpus, includes 570k sentence pairs (which we use as s_1 and s_2) labeled for three entailment classes: "entailment", "contradiction", and "neutrality"; each pair is enriched with 3 human-written, natural language explanations (Camburu et al., 2018), which we use in concatenation as our "gold" explanations.

---

[3] An example of how each dataset is preprocessed in the GEISER pipeline is provided in the Appendix.

StrategyQA (Multi-hop Question Answering). A question-answering dataset designed to require multiple-step strategic reasoning and/or implicit knowledge to answer a question. The dataset (Geva et al., 2021) comprises 2,780 strategy questions (which we use as s_2) with answer "yes" or "no" (labels), its decomposition into multi-step reasoning paths (which we use in combination as gold explanations) and evidence paragraphs giving the context of the question (which we use as s_1).

e-CARE (Causality). A dataset focused on causal reasoning, featuring human-annotated explanations for the causal questions, The dataset consists of 21k causal reasoning questions with both correct and incorrect answers (Du et al., 2022). We accommodate this dataset into our experimental setup by pairing both input sentences as s_1 and, for each pair, ask the question represented by s_2, focusing on whether the first sentence is the cause of the second (label "yes") or not (label "no").

e-RTE-3-it (Recognizing Textual Entailment in Italian). A dataset in Italian for Recognizing Textual Entailment (RTE), featuring pairs of texts-hypotheses and human-written, manually curated explanations for the entailment relation (Zaninello et al., 2023). It consists of 1,600 sentence pairs (which we use as s_1 and s_2, respectively) and is annotated with the same labels as e-SNLI.

5 Explanation Types

We test two different modes of explanation generation: explanations that *confirm* the given relationship between s_1 and s_2, explaining why it holds (*why*, *gold*) and explanations that potentially *contradict* the relationship between s_1 and s_2 explaining the circumstances when the relationship may not hold (*counterfactual*).

Why explanations. This kind of machine-generated explanation (`why`) is the most typical way to provide an explanation, i.e., the answer to a "why" question. In our setting, a why explanation is an answer to the question "*Why is r_c the relation holding between s_1 and s_2?*".

Gold explanations. These explanations (`gold`) are the explanations provided in the original dataset, either directly written or manually checked by humans given the correct relation r_c, thus falling into the label-confirming explanation type like *why* explanations. While the quality of human-generated explanations is generally considered high (e.g., we expect that they point out relevant and implicit information), there is no guarantee that, when used by a model M_2, they will perform better than model-generated ones. Therefore, for the purposes of this study, we evaluate them along with the generated ones and take them as a strong baseline, rather than consider them a target or reference explanation.

Counterfactual explanations. In our setting, a counterfactual (`cf`) explanation (Wachter et al., 2017; Verma et al., 2022) explicitly contradicts the golden label. It originates from the following question: "*What are the conditions in which relation r_c may not hold for s_1 and s_2?*". The aim of these explanations is to test the robustness of models to potentially false or misleading information, as well as highlight how different models may be differently sensitive to explanation injection[4].

6 Experiments

6.1 Experimental Setup

Models. We utilize three open-access language models of comparable size, which we combine as both M_1 and M_2: Llama-3-8B-Instruct (Team Llama et al., 2024), Gemma-7b-it (Gemma et al., 2024) and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025; Qwen et al., 2025).

To compute Cosine Similarity (Section 3.3), we use *sentence-transformers/all-MiniLM-L6-v2* (Wang et al., 2020). For Entailment, we use the pre-trained NLI model *deberta-large* (Liu et al., 2019), fine-tuned on the Multi-Genre NLI dataset (Williams et al., 2018).

Prompting and Inference Details. Our implementation leverages the HuggingFace's `lm_eval harness` library to ensure consistent and reproducible evaluation across tasks. For M_1 generation, we use the output type *generate_until*. We employ greedy decoding for all experiments, and all prompts are constructed in English (so that all explanations are returned in English, regardless of input)[5]. For M_2 prediction, we use the *multiple_choice* output type, which calculates logits for a given set of labels.

---

[4] See prompts and example explanations in the Appendix.

[5] Due to computational constraints, we used the first 800 examples from the test sets of each dataset to keep generation within our capacity limits. This approach allowed us to maintain a balance between comprehensive evaluation and practical feasibility.

	noexp	**gold**	**cf-llama-m1**	**cf-gemma-m1**	**cf-deepsk-m1**	**why-llama-m1**	**why-gemma-m1**	**why-deepsk-m1**
ESNLI (3 labels)								
llama-m2	0.54	0.71	0.34	0.44	0.59	0.72	0.89	**0.95**
gemma-m2	0.61	0.79	0.53	0.46	0.61	0.76	0.90	**0.95**
deepseek-m2	0.34	0.34	0.40	0.36	0.61	0.45	0.70	**0.96**
all-m2	0.50	0.61	0.42	0.42	0.60	0.64	0.83	**0.95**
SQA (2 labels)								
llama-m2	0.64	0.78	0.62	0.66	0.46	0.80	0.75	**0.91**
gemma-m2	0.62	0.68	0.55	0.58	0.50	0.81	0.75	**0.88**
deepseek-m2	0.45	0.45	0.43	0.38	0.52	0.45	0.45	**0.77**
all-m2	0.57	0.64	0.53	0.54	0.49	0.69	0.65	**0.85**
ECARE (2 labels)								
llama-m2	0.53	0.56	0.51	0.54	0.61	0.81	0.76	**0.91**
gemma-m2	0.48	0.71	0.54	0.51	0.62	0.92	0.75	**0.94**
deepseek-m2	0.48	0.49	0.50	0.48	0.59	0.53	0.54	**0.83**
all-m2	0.50	0.59	0.52	0.51	0.61	0.75	0.68	**0.89**
ERTEIT (3 labels)								
llama-m2	0.48	0.53	0.29	0.26	0.20	**0.70**	0.62	0.67
gemma-m2	0.44	0.49	0.23	0.19	0.20	**0.70**	0.59	0.67
deepseek-m2	0.48	0.48	0.50	0.38	0.20	0.58	0.53	**0.67**
all-m2	0.47	0.50	0.34	0.28	0.20	0.66	0.58	**0.67**

Table 2: Accuracy of M_2 models across the four datasets and explanation types, using explanations generated by M_1. Explanations marked as *noexp* and *gold* represent the baselines. Values are reported as accuracy scores of each M_2 model and as mean across all M_2 models (*all-m2*), with standard errors omitted for brevity. The best-performing explanation type for each M_2 is boldfaced.

To make generated explanations comparable to gold explanations, we ask M_1 to explain in approx. 3 sentences. To include the explanations in Step 2, we prompt M_2 to use a "hint" to give its answer, represented by the explanation.

Anonymization to Prevent Label Leakage. To ensure that the explanations do not simply suggest the right answer without genuinely being informative, we "anonymize" them by substituting each explicit reference to the labels with a placeholder (`XXX`) using regular expressions to fetch either the label (e.g. "YES" and "NO") or words directly connected to the relation (e.g. "contradict", "contradiction" etc.). Moreover, we explicitly ask the M_1 model to avoid stating the answer directly when generating the explanation.

6.2 Evaluation

Baselines. We select two baselines: no explanation given (**`noexp`**), where the model M_2 performs 0-shot relation r_p prediction; human explanation (**`gold`**), where we use the explanation provided in the original dataset as the hint, providing a strong baseline. Gold explanations too, like the generated ones, underwent the process of anonymisation.

Explanation quality. For explanation quality through GEISER, we calculate the average accuracy (**`acc`**) of each $M2$ model separately using either the explanations generated by the same model ($M_1 = M_2$), or by another model ($M_1 \neq M_2$). We report average accuracy for each explanation type/M_1, and both separately and ensembling by M_2, along with the accuracy obtained by with gold and noexp baselines (Table 2).

Correlation with Implicitness. To study the correlation of the selected implicitness measures (CS, COS, NLI) with explanation quality, we report the average score separately for each explanation type/M_1, each ranging from 0 to 1 (Table 3. Then, we calculate the Pearson correlation coefficient (r) ($p = 0.05$), to assess the linear relationship between each M_2 accuracy for all explanation types in each dataset, and the association measure for the same system for that dataset.

7 Results and Discussion

Accuracy Trends. In Table 2 we report the performances on the GEISER experiments for the four datasets under different explanation types, both with $M_1 = M_2$ and $M_1 \neq M_2$. Results are re-

	noexp	gold	cf-llama-m1	cf-gemma-m1	cf-deepsk-m1	why-llama-m1	why-gemma-m1	why-deepsk-m1
ESNLI (3 labels)								
Causal Strength	0.45	0.69	0.48	0.51	0.65	0.61	0.67	0.81
Cosine Similarity	0.05	0.70	0.48	0.46	0.48	0.44	0.56	0.47
Entailment prob.	0.34	0.42	0.12	0.13	0.82	0.22	0.23	0.77
SQA (2 labels)								
Causal Strength	0.55	0.70	0.53	0.58	0.71	0.64	0.69	0.77
Cosine Similarity	0.05	0.58	0.75	0.78	0.69	0.75	0.83	0.76
Entailment prob.	0.08	0.03	0.07	0.08	0.62	0.10	0.17	0.89
ECARE (2 labels)								
Causal Strength	0.50	0.85	0.52	0.53	0.61	0.65	0.64	0.73
Cosine Similarity	0.06	0.46	0.55	0.51	0.44	0.61	0.63	0.45
Entailment prob.	0.44	0.20	0.17	0.13	0.81	0.29	0.19	0.88
ERTEIT (3 labels)								
Causal Strength	0.60	0.71	0.47	0.53	0.53	0.72	0.70	0.80
Cosine Similarity	0.05	0.75	0.44	0.48	0.39	0.45	0.63	0.46
Entailment prob.	0.58	0.38	0.29	0.27	0.87	0.50	0.45	0.93

Table 3: Mean association measures (Causal Strength, Cosine Similarity, Entailment probability) across datasets, models, and explanation types. The *noexp* and *gold* columns indicate the baselines results using no-explanation and the human-generated ones. The remaining columns indicate results obtained by counterfactual *cf* or *why* explanations generated by the three LLMs as M_1.

ported both separately for the different M_2 models, as well as the average accuracy across all models (*all-m2*).

The best scoring task was ESNLI with *why* explanations written by Deepseek, which also presents the largest gain over the *noexp* baseline (from 0.34 to 0.96 with Deepseek both as $M1$ and M_2). Despite presenting a very similar task, the lower scoring dataset and the smallest gains were with ERTEIT (min. 0.20 with *cf*, max. 0.70 with LLama's *why*). This seems to indicate that the models still struggle with languages other than English, or are possibly mislead by the language shift between the input (Italian) and the explanations (English).

The preferred explanations were those of Deepseek with all M_2s in all tasks, with the exception of ERTEIT, where Llama-m2 and Gemma-m2 scored higher with Llama-m1's *why* explanations.

Label-confirming explanations (*why*) consistently led to the highest accuracy across all datasets and models, confirming that explanations aligned with the gold label can meaningfully support the m2 model's decision-making. On the other hand, label-contradicting explanations generally scored lower than the *noexp* baseline, as was expected, indicating that "bad" explanations can indeed be detrimental to the model's accuracy. However, there are a few cases where *cf* explanations improved over the *noexp* baseline, specifically on ESNLI and ECARE and mainly with Deepseek-m2's explanations. We manually inspected a sample of the *cf* explanations that led to a correct prediction, and noticed a common trend: in fact, in these cases either the model "refused" to support the opposite label, or it produced a long "chain of thought" style explanation which was truncated, and therefore did not contain the section of the explanation supporting the opposite label. This was especially the case with Deepseek, which produced an initial "reasoning" independently of the supported relationship, which was helpful for the downstream model to predict the correct label. Finally, it is worth noticing that machine-generated *why* explanations consistently outperform the human-generated ones (*gold*), which nonetheless are beneficial to prediction accuracy compared to *noexp*.

Association Measures. We computed three association measures between the explanation and the input: causal strength, cosine similarity, and entailment probability (Table 3). Label-confirming explanations showed higher values on all three measures compared to other conditions. For example, in SQA, the average causal strength and cosine similarity were highest (0.64 and 0.75, respectively) for label-confirming explanations, indicating a stronger semantic and causal link to the input. This confirms that these metrics are not only sensitive to novelty but also to relevance (*cf* being new information, which is not relevant for the correct label). On the other hand, entailment probability showed more variation across datasets, likely due to inherent differences in how entailment is interpreted in each task.

Correlations with Accuracy. To identify which measures best predict explanation utility, we correlated the association measures with accuracy across all explanation types (*gold*, *cf*, *why*) and generative models (Table 4). While all measures generally show a positive correlation with accuracy, Causal strength showed the highest and most significant correlation across all datasets (e.g., 0.86 for llama-m2 in ESNLI, 0.90 for all-m2 in ERTEIT). This suggests that the extent to which an explanation causally supports the input is a reliable predictor of its usefulness. Cosine similarity and entailment probability were weaker and less consistent predictors, though entailment reached high correlations in specific cases (e.g., 0.91 for deepseek-m2 on SQA).

Implications. These results suggest that injecting label-confirming explanations improves model performance, particularly when the explanations exhibit strong causal links to the input. Among the evaluated association measures, causal strength emerges as the most promising indicator of explanation quality. This highlights its potential as a diagnostic tool for filtering or scoring explanations before injection. Cosine similarity and entailment probability offer additional, though less robust, signals.

	CS	**COS**	**NLI**
ESNLI			
llama-m2	**0.86**	0.29	0.45
gemma-m2	0.33	0.33	0.33
deepseek-m2	0.63	0.19	0.63
all-m2	0.68	0.28	0.53
SQA			
llama-m2	0.42	0.21	0.14
gemma-m2	0.54	0.21	0.21
deepseek-m2	**0.71**	0.13	**0.91**
all-m2	0.62	0.21	0.48
ECARE			
llama-m2	0.43	0.37	0.46
gemma-m2	0.66	0.47	0.47
deepseek-m2	0.36	0.08	**0.80**
all-m2	0.54	0.36	0.53
ERTEIT			
llama-m2	**0.92**	0.20	0.18
gemma-m2	**0.93**	0.18	0.18
deepseek-m2	**0.70**	0.13	-0.02
all-m2	**0.90**	0.18	0.17

Table 4: Pearson correlation coefficient (r), between M_2 accuracy across the four datasets and the three association measures: Causal Strength (CS), Cosine Similarity (COS) and Entailment probability (NLI). The boldfaced figures indicate statistical significance according to a t-test with $n-2$ degrees of freedom, and $p = 0.05$.

8 Conclusion

In this study, we tested the effects of explanations on LLMs, showing that they can significantly improve their accuracy in predicting relations between sentences. This improvement is consistent across different models, datasets, and explanation types. Our experiments also show a correlation between explanation effectiveness and the degree of implicit knowledge conveyed by the explanations, suggesting that explanations that introduce novel and relevant information are more likely to be helpful to LLMs. Furthermore, our analysis reveals that different LLMs exhibit varying sensitivity to different explanation types. Our findings contribute to research on the role of explanations in enhancing LLM performance. By understanding the nuances of model sensitivity to different explanation types and the ways in which explanations contribute to implicit knowledge acquisition, we can develop more effective techniques for explaining and improving the reasoning capabilities of LLMs. Future work should explore how to automatically generate or filter explanations with high causal alignment to further boost downstream model performance.

Limitations

We focus on a specific type of NLP task involving implicit knowledge and investigate the impact of explanations on relation prediction. Further research is needed to extend these findings to a broader range of NLP tasks and model architectures.

Our measurement of implicitness relies on metrics like cosine nli and casual strength, which do not distinguish between relevance and novelty, and may not fully capture the nuanced nature of implicit knowledge in language. Finer-grained techniques are needed for a comprehensive evaluation of implicitness. Future work should explore additional features, such as explanation length and syntactic complexity, to better understand their interplay with model performance.

We utilized a controlled experimental setup, where explanations are provided in a specific format and injected into the model during inference. Real-world applications might involve more complex scenarios with less controlled input and output formats.

Also, while our study focused on sentence-pair tasks, the GEISER pipeline can in principle be extended to multi-hop reasoning chains and other explanation-rich settings by iteratively injecting intermediate explanations, which we plan to explore in future work.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2011. The seventh pascal recognizing textual entailment challenge. *Theory and Applications of Categories*.

Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. 2023. A survey on xai and natural language explanations. *Information Processing Management*, 60(1):103111.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Shaobo Cui, Lazar Milikic, Yiyang Feng, Mete Ismayilzada, Debjit Paul, Antoine Bosselut, and Boi Faltings. 2024. Exploring defeasibility in causal reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6433–6452.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-CARE: a new dataset for exploring explainable causal reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 432–446, Dublin, Ireland. Association for Computational Linguistics.

Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, et al. 2024. Team gemma and : Open models based on gemini research and technology.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Tirthankar Ghosal, Tanik Saikh, Tameesh Biswas, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Novelty detection: A perspective from natural language processing. *Computational Linguistics*, 48(1):77–117.

Tirthankar Ghosal, Amitra Salam, Swati Tiwari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. TAP-DLND 1.0 : A corpus for document level novelty detection. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mareike Hartmann and Daniel Sonntag. 2022. A survey on improving NLP models with human explanations. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 40–47, Dublin, Ireland. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2024. Using natural language explanations to improve robustness of in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13477–13499, Bangkok, Thailand. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Alexander Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. 2016. What's in an explanation? characterizing knowledge and inference requirements for elementary science exams. In *International Conference on Computational Linguistics*.

Masahiro Kaneko and Naoaki Okazaki. 2023. Controlled generation with prompt insertion for natural language explanations in grammatical error correction.

Been Kim, Rajiv Khanna, and Oluwasanmi O. Koyejo. 2016. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. *ArXiv*, abs/1606.04155.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

James L. McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. 2020. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences*, 117(42):25966–25974.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4179–4192, Online. Association for Computational Linguistics.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2022. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Lin CY ROUGE. 2004. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 887–890. ACM.

AI@Meta Team Llama, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. 2024. The llama 3 herd of models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. 2022. Counterfactual explanations and algorithmic recourses for machine learning: A review.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2).

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2020. Measuring association between labels and free-text rationales. In *Conference on Empirical Methods in Natural Language Processing*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, Samuel R. Bowman, Martin Abadi, and Antoine Bordes. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *Transactions of the Association for Computational Linguistics*, 6:309–324.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30378–30392. Curran Associates, Inc.

Andrea Zaninello, Sofia Brenna, and Bernardo Magnini. 2023. Textual entailment with natural language explanations: The italian e-rte-3 dataset.

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

Appendix

Illustrative Examples from the Four Datasets

Example 1 — ESNLI (with `gold` explanation)

Task Input

Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.
Hypothesis: The church is filled with song.
Label: entailment

Prediction Prompt (with `gold` explanation)

Your task is to predict the entailment relationship (entailment, neutral, contradiction) between a premise and a hypothesis given a hint.
Premise: This church choir sings to the masses as they sing joyous songs from the book at a church. Hypothesis: The church is filled with song. Hint: "Filled with song" is a rephrasing of "choir sings to the masses." Hearing song brings joyousness in the church. If the choir sings, then the church is filled with song. Answer:

Example 2 — SQA (with `noexp` explanation)

Task Input

Context: The Police = English rock band.
Question: Could the members of The Police perform lawful arrests?
Label: no

Prediction Prompt (with `noexp` explanation)

Your task is to answer a question with ’yes’ or ’no’ given a context and a hint.
Context: The Police = English rock band. Question: Could the members of The Police perform lawful arrests? Hint: None. Answer:

Example 3 — ECARE (with `why` explanation)

Task Input

Sentence 1: The woman gave birth to a child.
Sentence 2: The child brought psycho-physical phenomena on a new life.
Label: yes

Generation Prompt

Your task is to explain in approx. 3 sentences the answer to a question given a pair of sentences, without stating the answer directly.
Sentence 1: The woman gave birth to a child. Sentence 2: The child brought psycho-physical phenomena on a new life. Question: Is sentence 1 the cause of sentence 2? Answer = yes Explanation (max 3 sentences, avoid stating the answer directly):

Prediction Prompt (with `why` explanation by *DeepSeek*)

Your task is to answer a question with 'yes' or 'no' given a pair of sentences and a hint.
Sentence 1: The woman gave birth to a child. Sentence 2: The child brought psycho-physical phenomena on a new life. Question: Is sentence 1 the cause of sentence 2? Hint: Sentence 1 describes an event that happened before Sentence 2. In many cases, the XXX of an event is the immediate preceding event. Additionally, if Sentence 1 directly leads to Sentence 2, such as in the scenario where the woman gave birth to a child, it can be inferred that Sentence 1 is the XXX of Sentence 2. Answer:

Example 4 — ERTEIT (with `cf` explanation)

Task Input

Premise: Claude Chabrol (nato il 24 giugno 1930) è un regista francese diventato famoso negli ultimi 40 anni sin dal suo primo film, *Le Beau Serge*, per le sue agghiaccianti storie di omicidi, tra cui *Le Boucher*.
Hypothesis: *Le Beau Serge* è stato diretto da Chabrol.
Label: entailment

Generation Prompt

Your task is to convince me in approx. 3 sentences that the entailment relationship between a premise and a hypothesis is WRONG, without stating the relationship directly.
Premise: Claude Chabrol [...] *Le Beau Serge* [...] Hypothesis: *Le Beau Serge* è stato diretto da Chabrol. Entailment relationship: entailment. Explanation (max 3 sentences, avoid stating the relationship directly):

Prediction Prompt (with `cf` explanation by *Gemma*)

Your task is to predict the entailment relationship (entailment, neutral, contradiction) between a premise and a hypothesis given a hint.
Premise: Claude Chabrol [...] *Le Beau Serge* [...] Hypothesis: *Le Beau Serge* è stato diretto da Chabrol. Hint: The premise describes a renowned French director, Claude Chabrol, known for his intense murder stories. The hypothesis assumes that Chabrol directed the film *Le Beau Serge*. However, the premise does not necessarily XXX the hypothesis, as it does not provide any information about the film's authorship. Therefore, the XXX relationship between the premise and the hypothesis is incorrect. Answer:

Latent Traits and Cross-Task Transfer: Deconstructing Dataset Interactions in LLM Fine-tuning

Shambhavi Krishna[1,*] **Atharva Naik**[1,*] **Chaitali Agarwal**[1,*]
Sudharshan Govindan[1,*] **Haw-Shiuan Chang**[1,†] **Taesung Lee**[2,‡]
[*]Equal contribution
[1]University of Massachusetts Amherst [2]Meta
`{shambhavikri,atharvashrik,cragarwal,sgovindan}@umass.edu`
`hschang@cs.umass.edu, elca4u@gmail.com`

Abstract

Large language models are increasingly deployed across diverse applications. This often includes tasks LLMs have not encountered during training. This implies that enumerating and obtaining the high-quality training data for all tasks is infeasible. Thus, we often need to rely on transfer learning using datasets with different characteristics, and anticipate out-of-distribution requests. Motivated by this practical need, we propose an analysis framework, building a transfer learning matrix and dimensionality reduction, to dissect these cross-task interactions. We train and analyze 10 models to identify latent abilities (e.g., Reasoning, Sentiment Classification, NLU, Arithmetic) and discover the side effects of the transfer learning. Our findings reveal that performance improvements often defy explanations based on surface-level dataset similarity or source data quality. Instead, hidden statistical factors of the source dataset, such as class distribution and generation length proclivities, alongside specific linguistic features, are actually more influential. This work offers insights into the complex dynamics of transfer learning, paving the way for more predictable and effective LLM adaptation.

1 Introduction

Large Language Models (LLMs) demonstrate remarkable capabilities across diverse tasks, yet their deployment in real-world applications faces significant practical constraints. Cost and latency considerations render giant all-purpose models impractical for many use cases, driving widespread adoption of task-specific fine-tuning. However, this approach encounters a fundamental challenge: high-quality training data for target tasks is often unavailable or proprietary. Moreover, deployed LLMs routinely face out-of-distribution (OOD) requests that extend beyond their fine-tuning scope. This is especially true for agentic systems, which rely heavily on cross-domain skill transfer to perform diverse sequences of tasks. These realities necessitate a deeper understanding of transfer learning.

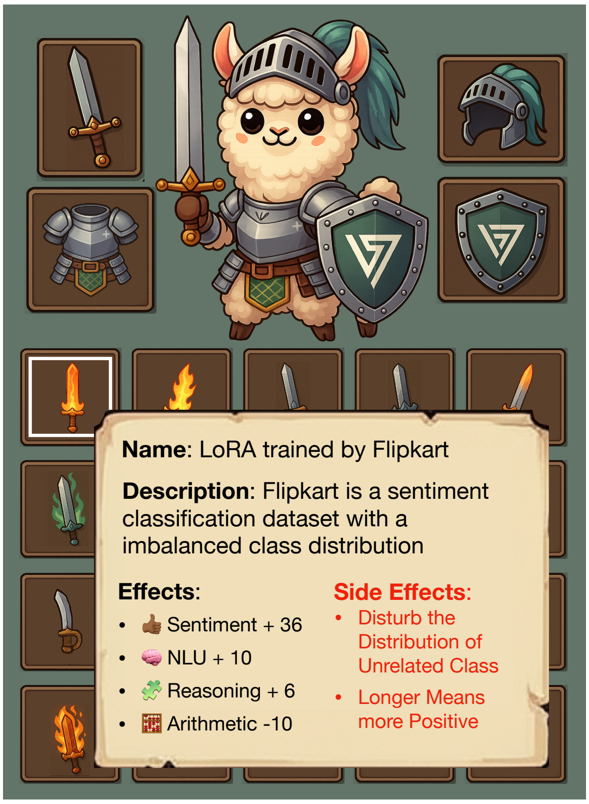

Name: LoRA trained by Flipkart
Description: Flipkart is a sentiment classification dataset with a imbalanced class distribution
Effects:
Sentiment + 36
NLU + 10
Reasoning + 6
Arithmetic -10
Side Effects:
Disturb the Distribution of Unrelated Class
Longer Means more Positive

Figure 1: Illustration of our motivations. LLMs such as Llama can be equipped with many different performance enhancers such as LoRA fine-tuned on a specific dataset. Our goal is to discover the potential impacts on out-of-domain tasks and side effects of each equipment.

Traditional transfer learning research has pri-

[†] Corresponding author.

[‡]This work was partly done when the author was at Anthropic.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 225–241
November 8-9, 2025 ©2025 Association for Computational Linguistics

marily focused on scenarios where source and target tasks share the same domain, assuming that similar or related in-domain data provides useful signal for the target task. However, the diverse task landscape that modern LLMs navigate demands deeper understanding of OOD transfer effects. Our experiments using Low-Rank Adaptation (LoRA) reveal counterintuitive transfer behaviors: fine-tuning on one dataset can yield surprising performance improvements or degradations on seemingly unrelated tasks, often defying expectations based on semantic similarity (illustrated conceptually in Figure 1). This unpredictability creates significant challenges for practitioners selecting optimal source datasets for fine-tuning, particularly in resource-constrained environments where training efficiency is paramount, or when acquiring pre-trained LoRA adapters from service providers without clear transferability guarantees.

In this paper, we propose a framework to analyze how the source fine-tuning dataset influences the performances on the target datasets in transfer learning and use this framework to systematically characterize the OOD generalization of an LLM using multiple LoRA adapters. Our analysis framework first constructs a performance matrix across different source and target tasks. We apply Principal Component Analysis (PCA) to this matrix to uncover latent abilities or "traits" that fine-tuned LLMs acquire from the transfer learning. We demonstrate that straightforward factors like source data quality or simple source-target similarity often fail to explain observed transfer learning effects. Instead, we highlight the critical role of more subtle, "hidden" statistical properties of the source training data (e.g., output length distribution, label imbalance) and learned sensitivities to specific linguistic features.

Our work aims to provide actionable insights into the selection of the source dataset for fine-tuning, fostering a deeper understanding of the interactions among the datasets and guiding the development of more robust LLM adaptation strategies. In our experiments, we fine-tune the Llama 3.2 3B base model (Dubey et al., 2024) using LoRA and systematically evaluate models fine-tuned on one dataset across datasets for math, coding, natural language inference, sentiment, and toxicity detection tasks to map diverse data interactions. Through analyzing the fine-tuned LLM and datasets, we observe several surprising cross-domain interactions, including: (1) the impact of source data generation length on fine-tuned model outputs; (2) asymmetric enhancement through out-of-domain fine-tuning datasets; and (3) the profound effects of source label imbalance on both in-domain and OOD performance.

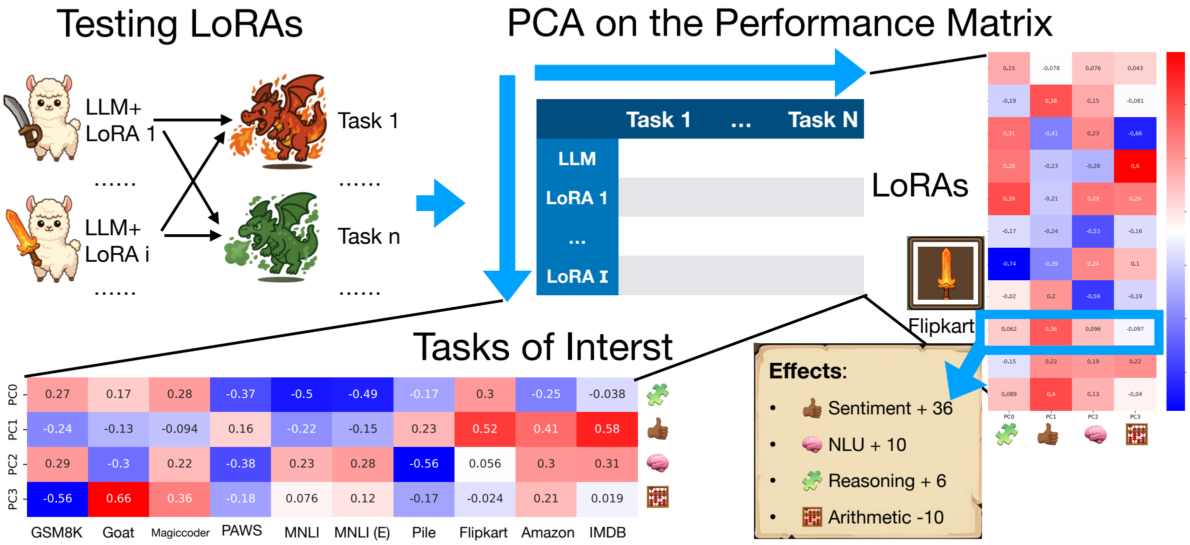
2 Related Work

The transfer learning of fine-tuning language models is investigated by several existing works (Vu et al., 2020; Chang and Lu, 2021; Parvez and Chang, 2021; Weller et al., 2022; Padmakumar et al., 2022; Li et al., 2024b; Schulte et al., 2024; Yang et al., 2024; Li et al., 2024a). Most studies focus on identifying similar tasks for positive transfer effect through fully fine-tuning small language models. Instead, our work focuses on modeling the impact of LoRA fine-tuning and discovering the often-overlooked side effect of the source training datasets including out-of-domain and out-of-distribution datasets. Compared to the full fine-tuning, Biderman et al. (2024); Ghosh et al. (2024) find LoRA “learns less and forgets less”, which potentially preserves out-of-domain base model capabilities better. This is one of the main reasons behind LoRA’s effectiveness and popularity. Nevertheless, we demonstrate that LoRAs, which are fine-tuned on many source datasets, could still cause several types of undesirable side effects when being evaluated on a wide range of target tasks.

Methodologically, our analysis framework is related to Ruan et al. (2024), which employs PCA to analyze observational scaling laws and the predictability of LLM performance across different model sizes and tasks. Some recent findings also support our discoveries of hidden factors. For example, Zhang et al. (2025) report that instruction fine-tuning with coding data can sometimes negatively impact mathematical reasoning. Guha et al. (2025) find that the length distribution of the instruction tuning training data could affect the LLMs’ code generation ability. Min et al. (2022); Kung and Peng (2023); Guha et al. (2025) discover that the format of the fine-tuning data might be more important than its content or correctness. Our work confirms their findings and provides a more comprehensive list of latent traits that influence LoRAs’ performance.

3 Methodology

In our framework, we first prepare N representative tasks/datasets of interest and fine-tune LLMs



	GSM8K	Goat	Magiccoder	PAWS	MNLI	MNLI (E)	Pile	Flipkart	Amazon	IMDB
PC0	0.27	0.17	0.28	-0.37	-0.5	-0.49	-0.17	0.3	-0.25	-0.038
PC1	-0.24	-0.13	-0.094	0.16	-0.22	-0.15	0.23	0.52	0.41	0.58
PC2	0.29	-0.3	0.22	-0.38	0.23	0.28	-0.56	0.056	0.3	0.31
PC3	-0.56	0.66	0.36	-0.18	0.076	0.12	-0.17	-0.024	0.21	0.019

PC0	PC1	PC2	PC3
0.15	-0.078	0.076	0.043
-0.19	0.38	0.15	-0.081
0.31	-0.41	0.23	-0.66
0.28	-0.23	-0.28	0.6
0.39	-0.21	0.29	0.26
-0.17	-0.24	-0.53	-0.16
-0.74	-0.39	0.24	0.1
-0.02	0.2	-0.59	-0.19
0.062	0.36	0.096	-0.097
-0.15	0.22	0.19	0.22
0.089	0.4	0.13	-0.04

Figure 2: Discovering the latent traits of LoRA through PCA. The performance matrix stores the performance of I LoRAs on N tasks. A PCA factorizes the performance matrix into two matrices: the top four eigenvectors/bases in the bottom left and the linear weights that combine the eigenvectors/bases in the right. More red means the values are higher. Based on the eigenvectors, we identify the meaning of each principal component as our latent traits, and we can use the linear weights of the LoRA trained on Flipkart as its influence to the other datasets through the traits.

on these N datasets to acquire I fine-tuned LLM variants. In this paper, we use LoRA to fine-tune LLMs, but the framework could be applied to any fine-tuning method (e.g., full fine-tuning, prompt tuning (Lester et al., 2021), BitFit (Zaken et al., 2022), etc.) or any variants of LLMs (e.g., in-context learning or chain of thoughts).

Viewing each fine-tuned LoRA adapter as a specialized piece of equipment in a practitioner's toolkit, a crucial challenge is selecting the right tool for a new task. The conventional approach assumes a tool's effectiveness is dictated by its labeled domain, for instance, using a 'sentiment' adapter for a sentiment task. However, these tools might come with unexpected side-effects and hidden capabilities driven by the latent statistical properties of their training data, not just their domain.

To solve this issue, we evaluate the I fine-tuned LLMs on the N datasets on their accuracy, and organize the pairwise results into a $I \times N$ performance matrix as shown in Figure 2. Note that since our goal is to measure the impact on out-of-domain tasks, we focus more on relative gains, rather than the absolute performance numbers.

Throughout this paper, we denote the base LLM as M; a model fine-tuned on a dataset D as M["D"]; and the output performance of such a model on the evaluation data X is denoted M["D"](X). For example, M["Flipkart"](GSM8K) refers to the score of the model fine-tuned on the Flipkart dataset and tested on GSM8K. In the performance matrix, M["Flipkart"](GSM8K) corresponds to the row for Flipkart and column for GSM8K.

To understand the overall characteristics and transfer learning impact across these datasets, we decompose the performance matrix using PCA. Each principal component corresponds to a group and the tasks with high values in the corresponding eigenvector belong to the group. In this way, similar evaluation tasks whose LLM scores have high correlations will cluster together. We can then use the common attribute of the tasks in a group as its name - a standard practice to make the abstract mathematical components interpretable. Guided by the PCA results, we discover the transfer learning patterns among the tasks of interest and further investigate the outliers in the performance matrix. We then conduct analyses to identify the factors that could explain the patterns and outliers.

4 Experimental Setup

We curate a diverse set of datasets spanning mathematical reasoning (MetaMath (Yu et al., 2024), GSM8K (Cobbe et al., 2021), and Goat (Liu and Low, 2023)), code generation (Magicoder (ISE-UIUC, 2023)), Natural Language Inference (NLI) (PAWS (Zhang et al., 2019) and MNLI (Williams et al., 2018)), Sentiment analysis (Flipkart Sentiment (KayEe), Amazon Reviews (Zhang and Yassir, 2022), and IMDB Reviews (Maas et al., 2011)),

Fine-tuned on	**GSM8K**	**Goat**	**Magicoder**	**PAWS**	**MNLI**	**MNLI (E)**	**Pile**	**Flipkart**	**Amazon**	**IMDB**
None (Original LLM)	9.78	6.36	21.55	46.30	33.30	33.75	38.45	63.55	31.80	51.45
MetaMath	44.96	5.40	20.50	44.55	34.65	32.95	42.25	46.10	21.60	49.65
Goat	13.42	24.65	21.65	44.55	33.10	35.10	47.15	57.85	25.45	53.00
Magicoder	19.18	8.45	29.38	45.70	33.35	34.10	37.35	70.90	28.10	51.15
PAWS	8.57	7.75	20.75	70.05	34.90	33.15	49.05	12.10	21.85	46.80
MNLI (Eng.)	9.33	6.00	20.55	57.65	69.50	83.45	51.10	5.85	37.70	53.50
Pile	12.66	8.16	21.47	56.35	35.70	33.65	85.25	83.90	32.80	51.00
Flipkart	14.59	5.96	21.84	55.55	33.65	36.25	49.10	92.65	38.70	77.15
Amazon	12.97	9.15	22.48	55.45	39.10	38.35	47.90	39.95	61.25	69.05
IMDB	12.78	6.96	22.19	55.40	34.00	34.70	46.35	85.55	31.40	91.45

Table 1: Model fine-tuning and cross-task evaluation results (% Automatic Accuracy or Accuracy from LLM-as-a-Judge). Each model was fine-tuned on a single dataset (leftmost column) and evaluated across multiple target tasks (column headers). MNLI (E) refers to MNLI English.

and toxicity detection (Pile (Korbak, 2024)). For more information about the datasets refer to Table 6 in Appendix A.1.

We employ Low-Rank Adaptation (LoRA) with rank 64 to fine-tune the Llama 3.2 3B base model M on each source dataset to get a fine-tuned M[Dataset]. [1] For all tasks, we report the accuracy using LLM-as-a-Judge. [2] Specifically, we use Llama 3.3 70B Instruct (Dubey et al., 2024) to judge if the generated answers are the same as the ground truth answer (see Appendix B for prompt). For each dataset, 10,000 samples are randomly chosen from its training split, and 2,000 from the test split unless specified otherwise. Model training specifics are detailed in Section A.2.

5 Results and Analyses

In this section, we show the analysis on how the statistical properties drive transfer learning regardless of the domain similarity. We show various statistical properties and their effects on performance for both in-domain transfer and out-of-domain transfer. The overall cross-task performance matrix is summarized in Table 1.

5.1 PCA Results

The results of PCA on the performance matrix are visualized in Figure 2. The first four eigenvectors, which explain around 75% of the total variance in the performance matrix, are presented at the bottom-left of the figure and each column corresponds to a target evaluation task in Table 1.

The first principal component (PC0) assigns positive values to GSM8K, Goat, Magicoder, and Flipkart, suggesting that PC0 measures the *reasoning* performance of LoRAs. Surprisingly, Flipkart is also included in the group. The second principal component (PC1) group consists of PAWS, Pile, Flipkart, Amazon, and IMDB, which are mostly *sentiment classification* datasets except for PAWS. The PC2 groups GSM8K, Magicoder, MNLI, MNLI (E), Amazon, and IMDB together, so we believe the group represents the general *natural language understanding* (NLU) performance. Finally, PC3 highlights the performance differences between GSM8K and Goat. Table 1 shows that LoRA fine-tuned on MetaMath actually decreases the performance on Goat. We hypothesize that this is because Goat tests the arithmetic for large numbers while GSM8K only requires the arithmetic for small numbers. Thus, we annotate PC3 as LoRAs' ability of performing *arithmetic* for large numbers due to its large positive value to Goat. The positive values of Magicoder and Amazon might indicate that solving these tasks also require this arithmetic skill.

According to our annotation of every principal component, we can characterize LoRA fine-tuned by every source dataset based on the values projected to each principal component. For example, the table on the right side of Figure 2 shows that LoRA from the Flipkart sentiment classification task improves sentiment ability the most as expected. Besides, it also slightly improves the NLU and reasoning ability of LLMs while degrading the arithmetic performance.

5.2 Analyzing Side Effects of Cross-Task Transfer Systematically

To map the behaviors of transfer learning, we categorize them using the 2 × 2 table in Table 2. This

---

[1] MetaMath is designed for training, so we replace MetaMath with GSM8K in evaluation.

[2] While widely used for scalable evaluation, we acknowledge that the LLM-as-a-Judge method may introduce its own inherent biases, a potential limitation of our evaluation framework.

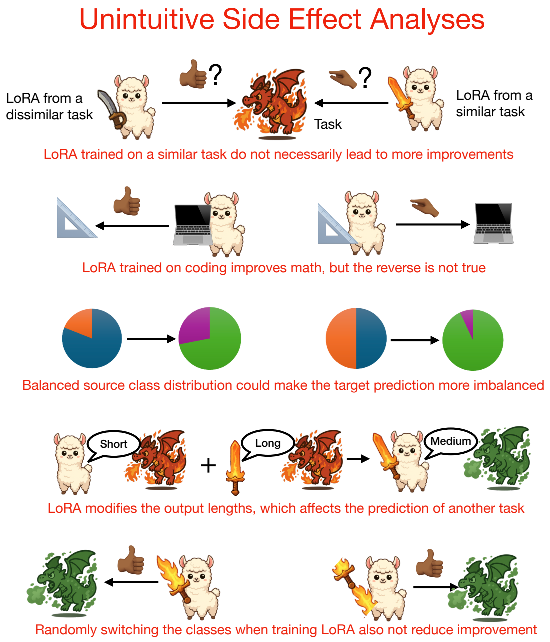

Unintuitive Side Effect Analyses
LoRA from a dissimilar task
?
?
Task
LoRA from a similar task
LoRA trained on a similar task do not necessarily lead to more improvements
LoRA trained on coding improves math, but the reverse is not true
Balanced source class distribution could make the target prediction more imbalanced
Short
Long
Medium
+
LoRA modifies the output lengths, which affects the prediction of another task
Randomly switching the classes when training LoRA also not reduce improvement

Figure 3: Unintuitive side effects of using LoRA adapters as specialized 'tools'. This figure illustrates surprising behaviors where a tool's performance is not predicted by its label: domain similarity can be misleading, skill transfer is often asymmetric, and hidden statistical properties like class balance and output length proclivities are transferred to new tasks with unexpected consequences.

table helps explain the counterintuitive results observed in our experiments: why a LoRA trained on the tasks from a different domain (e.g., a 'code generation' adapter) might surprisingly outperform an in-domain one for a specific mathematical task, or why two seemingly identical 'sentiment' LoRAs can have vastly different effects on the target task. The following sections will deconstruct the specific properties of these LoRAs, analyzing their generation length proclivities (Section 5.3), internal class distributions (Section 5.4), learned linguistic sensitivities (Section 5.5), and the correctness of the labels (Section 5.6) to explain the surprising dynamics. We illustrate the most notable side effects in Figure 3.

	Same Domain	**Different Domain**
Different Stats	**Unexpected Negative Transfer** • Amazon → Flipkart (both sentiment) shows poor transfer. • Flipkart (balanced vs. imbalanced) yields divergent results on other sentiment tasks.	**Asymmetric & Negative Transfer** • Asymmetric Transfer: Code → Math (+9.4) but Math → Code (-1.05). • Math → Sentiment shows strong negative transfer (e.g., Flipkart, -17.45).
Similar Stats	**Traditional Expectation** • IMDB → IMDB shows strong in-domain performance (91.45%). • MNLI → MNLI (E) is also strong (83.45%).	**Surprising Positive Transfer** • *Length Similarity:* Code → Math transfer outperforms in-domain Math → Math. • *Linguistic Transfer:* Classification → Math improves reasoning.

Table 2: A summary for cross-task side effects.

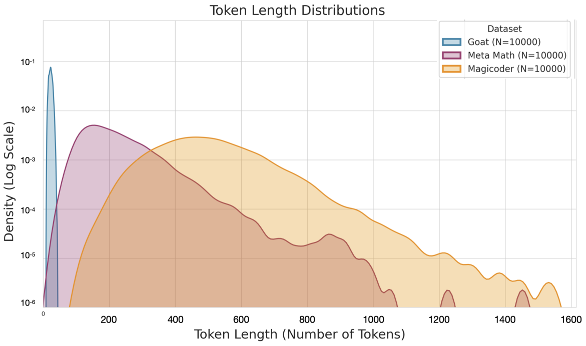


Figure 4: Generation length differences across Meta Math, Goat and Magicoder datasets.

5.3 Length Distribution

We observe that performance changes sometimes align with the length distributions of the fine-tuning and evaluation datasets, a characteristic learned by the model that influences output length on the target task. For example, while both Meta-Math/GSM8K and Goat are Math domain datasets, Goat has a significantly shorter generation length distribution (Figure 4). Fine-tuning on Magicoder, a code dataset with a length distribution more similar to Meta-Math/GSM8K's, proved more effective on GSM8K (+9.40 gain) than fine-tuning on the in-domain Goat dataset (+3.64 gain). This suggests that matching generation length proclivities can be crucial for positive transfer.

However, this phenomenon is sophisticated and influenced by several interacting factors (detailed in Appendix D):

- **Interpolation of Lengths:** Models fine-tuned on generation tasks often produce outputs whose lengths interpolate between the base model's tendencies and those of the fine-tuning data.
- **Classification Task Influence:** Fine-tuning on classification datasets generally preserves the base model's generation length on OOD generation tasks, unless the classification data itself has a strong length bias.
- **Dataset-Specific Length Transfer:** Certain

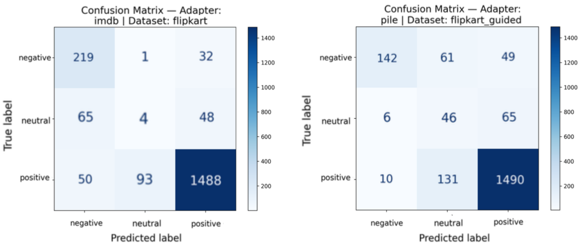


Figure 5: Confusion Matrices on Flipkart: M[IMDB] (left) vs. M[Pile] (right).

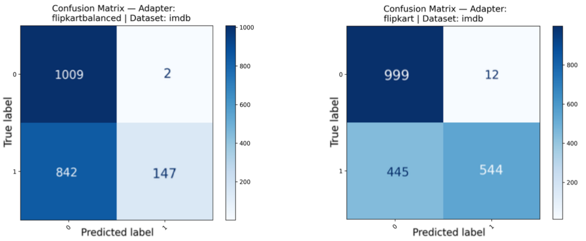


Figure 6: Confusion Matrices on IMDB (binary): M[Flipkart(Balanced)] (left) vs. M[Flipkart(Imbalanced)] (right) : 0=negative, 1=positive.

  datasets (e.g., Pile) can impart distinct length tendencies that transfer to OOD tasks.

- **Length Bias in Classification Inputs:** Correlations between input text length and class labels in a source classification dataset can be learned and transferred, affecting predictions on target classification tasks.

These findings suggest that generation length is a transferable latent trait. Models exhibit a form of "inertia", blending prior generation habits with newly learned ones from the fine-tuning data. This has implications for multi-task learning, as unintended output lengths could affect downstream performance or introduce subtle biases.

5.4 Class Distribution

In classification tasks, the model needs to learn the features of the input and predict a series of tokens representing a class. We find that fine-tuning can shift this output class distribution in unexpected ways for both in-domain and out-of-domain tasks. Notice that when analyzing the class distributions, we can often ignore the impact of the length distribution because the outputs of the classification tasks are typically only a couple of tokens.

Model (FT on)	Pile	Flipkart	Amazon	IMDB
Original LLM	38.45	63.55	31.80	51.45
Pile	85.25	83.90	32.80	51.00
Flipkart-imb.	39.10	92.65	38.70	77.15
Flipkart-bal.	50.05	N/A	38.80	57.80
Amazon	47.90	39.95	61.25	69.05
IMDB	46.35	85.55	31.40	91.45

Table 3: Label Imbalance Effects on Classification Tasks. Flipkart-bal. means Flipkart-balanced, Flipkart-imb, means Flipkart-imbalanced.

With the high similarity between the classification tasks, we could observe positive transfer between classification tasks for many dataset pairs (Table 3). For example, M[Pile](Flipkart) improves performance to 83.90%, and M[IMDB](Flipkart) improves to 85.55%, as compared to 63.55% from M(Flipkart). Moreover, we observed that prediction bias could be learned and applied to a different task, both in domain and across domains. For instance, Figure 5 shows that M[Pile] predicts 'neutral' more often on Flipkart than M[IMDB], which suggests that training on Pile (toxicity) might increase sensitivity to ambiguous language, while IMDB training (binary sentiment) pushes for definitive positive/negative calls.

To further isolate the effect of label distribution from the task itself, we increase the negative class ratio from around 20% to 50%. The newly created dataset is called Flipkart-balanced, while the original Flipkart is called Flipkart-imb. Comparing LoRA M[Flipkart-balanced] with M[Flipkart], Table 3 highlights target-dependent effects due to the class distribution similarity and the dissimilarity between the fine-tuning and evaluation datasets. M[Flipkart-balanced](Pile) performs better than M[Flipkart-imb.](Pile)(50.05% vs. 39.10%), while M[Flipkart-imb.](IMDB) is better (77.15% vs. 57.80%). Balancing may help tasks needing unbiased signals (toxicity - Pile), while natural imbalance can preserve useful priors for OOD tasks with similar distributions (sentiment - IMDB).

Figure 6 compares M[Flipkart-imb.](IMDB) and M[Flipkart-balanced](IMDB), which demonstrates a bias towards predicting 'negative', especially M[Flipkart-balanced]. This might be linked to learning spurious features like the input length and predicting long inputs as negative because negative reviews in Flipkart are longer than positive reviews, unlike IMDB's more uniform lengths as shown in Figure 7.

5.5 Transferring from Classification to Math

Fine-tuning on classification datasets shows a surprising ability to improve performance on mathe-

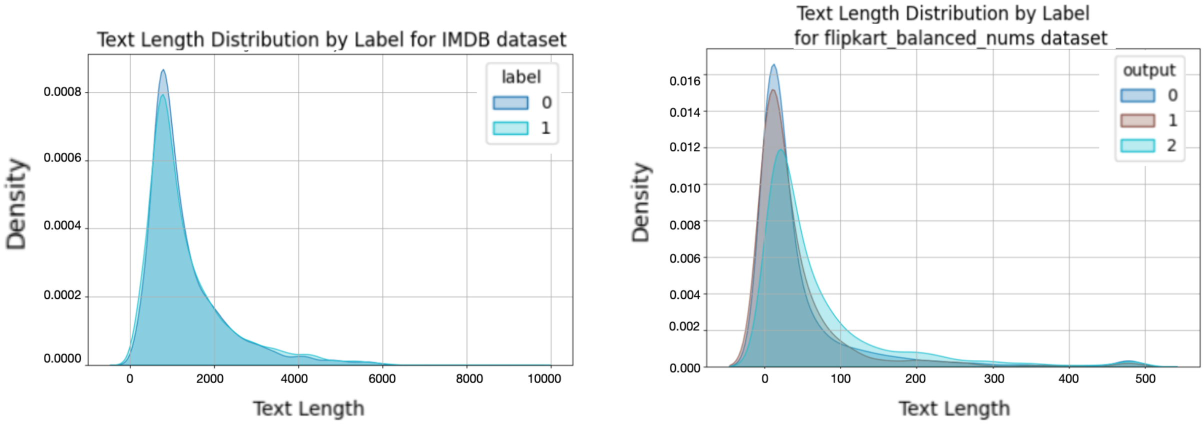
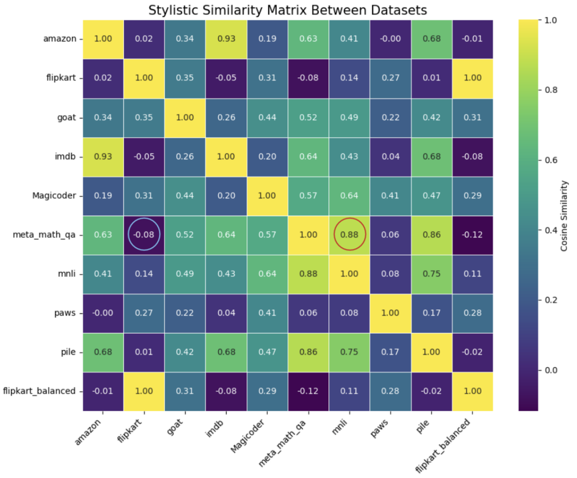


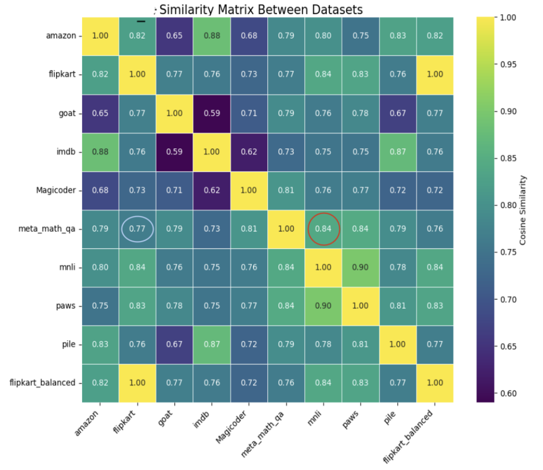
Figure 7: Text length distribution of each sentiment label from the kernel density estimation (KDE) for IMDB (left: 0=negative, 1=positive) and Flipkart (right: 0=positive, 1=neutral, 2=negative).

Model Fine-tuned on	GSM8K Acc. (%)	Goat Acc. (%)
None (Original LLM)	9.78	6.36
Flipkart (Imbalanced)	14.59	5.96
Flipkart (Balanced)	13.00	6.50
Amazon	12.97	9.15
IMDB	12.78	6.96
Pile	12.66	8.16
PAWS	8.57	7.75
MNLI (Eng.)	9.33	6.00

Table 4: Performance of models fine-tuned on classification datasets, evaluated on GSM8K and Goat.



(a) Stylistic Similarity Matrix Between Datasets.



(b) Semantic Similarity Matrix Between Datasets.

Figure 8: Stylistic Similarity (top) and Semantic Similarity (bottom) Matrices.

matical reasoning tasks, particularly GSM8K. Table 4 shows that several LoRAs trained on classification tasks improved GSM8K accuracy over the original LLM. For example, M[Flipkart] achieves 14.59%, which is much higher than 9.78 from M. This gain was less pronounced on Goat, a more arithmetic-focused dataset, suggesting the improvement is more related to linguistic reasoning than raw calculation.

One initial hypothesis was that overall stylistic similarity (Wegmann et al., 2022) or semantic similarity[3] between source classification datasets and target math datasets might predict transfer. However, these broad similarities did not consistently correlate with the observed improvements on GSM8K, indicating that more nuanced factors are at play. For example, Figure 8 highlights that MNLI has very high similarities with MetaMath while there is almost no positive transfer between them. In contrast, Flipkart could significantly improve GSM8K, while being stylistically very dissimilar to MetaMath.

Instead, our analysis (detailed in Appendix F) suggests that the improvement stems from an enhanced sensitivity to specific linguistic structures crucial for understanding and deconstructing word problems. Key observations include:

---

[3] MiniLM-L6-v2 from `https://www.sbert.net/`

- **Sensitivity to Syntactic Cues:** The fine-tuned LoRAs significantly improve the model's ability to interpret the grammatical structure of word problems, which is essential for translating text into correct mathematical operations. For example, Figure 9 shows that the model becomes better at identifying dependency relations like oprd (operand), which flags a number as an object to be acted upon, and parataxis, which links related clauses together. This enhanced syntactic proficiency is not just a linguistic improvement; it is the mechanism that allows the model to more reliably deconstruct complex sentences into accurate logical and mathematical steps. A failure to parse these cues correctly often leads to building the wrong equation (e.g., adding numbers that should be multiplied).

- **Asymmetric Transfer:** Our analysis revealed a significant asymmetric skill transfer. For example, classification datasets such as Flipkart, Amazon and IMDB improve the performance on GSM8K (+4.81, +3.19, +3.00, respectively), but training on MetaMath did not improve on Flipkart, Amazon and IMDB (-17.45, -10.20, -1.80). Similarly, a strong positive transfer from code to math was observed, where M[Magicoder] improved performance (+9.4) while the inverse resulted in -1.05.

- **Handling of Arithmetic Operations:** LoRAs fine-tuned on classification datasets lead to consistent gains in math reasoning, especially for high-frequency operations like addition, subtraction, and division (see Figure 17 in the appendix). This improvement appears to be linked to the model's increased ability to attend to relevant tokens and avoid premature termination, leading to more complete reasoning chains.

These findings highlight that fine-tuning on classification tasks can, perhaps counter-intuitively, refine a model's linguistic processing in ways beneficial for structured reasoning tasks like mathematics, beyond what simple dataset similarity would predict.

5.6 Importance of Labels

To further understand what drives performance, especially the linguistic understanding gained during fine-tuning, we investigated the direct role of labels.

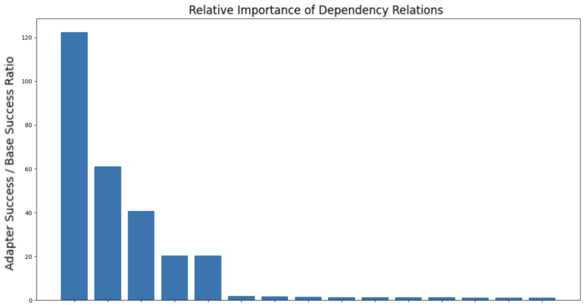


Figure 9: Performance ratio (LoRA success / original LLM success) on math problems. Gains on relations like oprd show the model's improved ability to parse grammatical structure is the key driver of its success.

Model (Fine-tuned on)	**Flipkart**	**IMDB**	**GSM8K**
M[Amazon]	15.85%	48.65%	3.34%
M[Amazon-mislabeled]	16.75%	48.80%	4.02%

Table 5: Impact of Fine-tuning on Wrong Labels Compared to Correct Labels (% Accuracy).

We tested if LoRAs are sensitive to incorrect labels. Interestingly, fine-tuning on Amazon reviews where labels are deliberately mislabeled yielded similar or even slightly better performance on OOD tasks like IMDB and GSM8K (Table 5), suggesting models can pick up underlying data structures even with noisy labels, or that the mislabeling process inadvertently created patterns beneficial for other tasks.

6 Conclusion

Our investigation into the cross-task dynamics of Low-Rank Adaptation (LoRA) confirms that transfer learning behavior is often unintuitive and defies explanations based on task domains or surface level dataset similarity. This work introduced a systematic framework to dissect these interactions, revealing that performance on target tasks is more influenced by the transfer of latent statistical and linguistic traits learned from a source dataset. We established the existence of these phenomena, such as asymmetric skill transfer and the impact of the class distribution, providing a new lens through which to view the fine-tuning process.

Our work paves the way for a more predictable and engineering-driven discipline around LLM adaptation. The logical next step is to move toward a modular approach, creating a "tool-belt" of skill adapters for agentic systems. An agent could then dynamically load a "conciseness adapter" for summarization or a "syntactic-parser adapter," like the one we observed emerging from classification data, for complex instruction understanding. While we demonstrated these dynamics on the Llama 3 architecture with LoRA, a critical next step is to validate and expand these findings across a broader range of models and adaptation methods to assess their scalability. By pursuing these avenues of modularity and prediction, we can build more robust and capable AI agents.

Acknowledgments

We are also deeply grateful to the valuable feedback from Mengxue Zhang, Wenlong Zhao, Dhruv Agarwal, and Prof. Andrew McCallum. This work was supported in part by Center for Data Science and in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Dan Biderman, Jacob P. Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. 2024. LoRA Learns Less and Forgets Less. *Trans. Mach. Learn. Res.*, 2024.

Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. Software available from wandb.com.

Ting-Yun Chang and Chi-Jen Lu. 2021. Rethinking Why Intermediate-Task Fine-Tuning Works. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S., Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. 2024. A Closer Look at the Limitations of Instruction Tuning. In *Forty-first International Conference on Machine Learning, ICML 2024*.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, and et al. 2025. OpenThoughts: Data Recipes for Reasoning Models. *Preprint*, arXiv:2506.04178.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

ISE-UIUC. 2023. Magicoder-oss-instruct-75k. `https://huggingface.co/datasets/ise-uiuc/Magicoder-OSS-Instruct-75K`. Accessed: 2025-03-09.

KayEe. Flipkart Sentiment Analysis. `https://huggingface.co/datasets/KayEe/flipkart_sentiment_analysis`. Accessed: 2025-04-27.

Tomek Korbak. 2024. Pile Toxicity Balanced. `https://huggingface.co/datasets/tomekkorbak/pile-toxicity-balanced`. Accessed: 2025-04-27.

Po-Nien Kung and Nanyun Peng. 2023. Do Models Really Learn to Follow Instructions? An Empirical Study of Instruction Tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023*. ACM.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Association for Computational Linguistics.

Dongyue Li, Aneesh Sharma, and Hongyang R. Zhang. 2024a. Scalable Multitask Learning Using Gradient-based Estimation of Task Affinity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024*. ACM.

Yingya Li, Timothy Miller, Steven Bethard, and Guergana Savova. 2024b. Identifying Task Groupings for Multi-Task Learning Using Pointwise V-Usable Information. *CoRR*, abs/2410.12774.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *Preprint*, arXiv:2305.14201.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. Association for Computational Linguistics.

Vishakh Padmakumar, Leonard Lausen, Miguel Ballesteros, Sheng Zha, He He, and George Karypis. 2022. Exploring the Role of Task Transferability in Large-Scale Multi-Task Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*. Association for Computational Linguistics.

Md. Rizwan Parvez and Kai-Wei Chang. 2021. Evaluating the Values of Sources in Transfer Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*. Association for Computational Linguistics.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Observational Scaling Laws and the Predictability of Language Model Performance. *CoRR*, abs/2405.10938.

David Schulte, Felix Hamborg, and Alan Akbik. 2024. Less is More: Parameter-Efficient Selection of Intermediate Tasks for Transfer Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*. Association for Computational Linguistics.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. *Preprint*, arXiv:2005.00770.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same Author or Just Same Topic? Towards Content-Independent Style Representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2022*. Association for Computational Linguistics.

Orion Weller, Kevin D. Seppi, and Matt Gardner. 2022. When to Use Multi-Task Learning vs Intermediate Fine-Tuning for Pre-Trained Encoder Transfer Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*. Association for Computational Linguistics.

Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng-Ann Heng, and Wai Lam. 2024. Unveiling the Generalization Power of Fine-Tuned Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2024*. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Xiang Zhang and Acharki Yassir. 2022. Amazon Reviews for SA fine-grained 5 classes.

Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. 2025. Unveiling the Impact of Coding Data Instruction Fine-Tuning on Large Language Models Reasoning. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*. AAAI Press.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*. Association for Computational Linguistics.

A Experimental Specifics

A.1 Dataset

Table 6 lists the datasets we considered for our analyses, ranging from generation to classification tasks, across different domains, class and length distributions, etc.

A.2 Implementation

We use the Llama 3.2 3B model (Dubey et al., 2024) as our base model. Fine-tuning is performed using LoRA (Hu et al., 2022) with a rank of 64 and an alpha of 32, applied to q_{proj}, k_{proj}, v_{proj}, o_{proj}, $gate_{proj}$, up_{proj}, and $down_{proj}$ layers. We use the AdamW optimizer with a cosine learning rate schedule. The Unsloth library (Daniel Han and team, 2023) is utilized for efficient training with gradient checkpointing. Experiments are tracked using Weights & Biases (W&B) (Biewald, 2020), and vLLM (Kwon et al., 2023) is used for optimized batch inference.

B LLM-as-a-Judge Prompt

The following prompt was used with Llama 3.3 70B Instruct to evaluate the model-generated outputs against the given ground truth for generation tasks. The LLM was instructed to provide a binary score (0 or 1) without explanations.

```
<|begin_of_text|><|start_header_id|>
system<|end_header_id|>

Your job is to check whether the AI's answer is correct.

Compare it with the correct answer and
score it as either 0
if the AI's answer is wrong or 1 if it is correct.

DO NOT provide any explanations.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
Correct Answer: {groundtruth_column}
AI Answer: {Model generated output}<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>

Score:
```

This prompt ensured a strict, explanation-free evaluation of the model's responses based on the provided ground truth.

C PCA Details

To increase the diversity of the LLM variants in the performance matrix before running PCA, we also test a few-shot baseline on Llama 3.2 1B base model. Its accuracy is 14.11 on GSM8K, 7.00 on Goat, 18.87 on Magicoder, 55.45 on PAWS, 40.35 on MNLI, 52.60 on MNLI (E), 52.55 on Pile, 90.95 on Flipkart, 48.35 on Amazon, 74.85 on IMDB. We use 5 shots for Amazon, 2 shots for Pile and IMDB, and 3 shots for the rest of the datasets.

The rows of the linear weight matrix on the right side of Figure 2 correspond to zero-shot Llama 3.2 3B base, few-shot Llama 3.2 1B base model, LoRA from GSM8K, LoRA from Goat, LoRA from Magicoder, LoRA from PAWS, LoRA from MNLI (E), LoRA from Pile, LoRA from Flipkart, LoRA from Amazon, LoRA from IMDB (from top to bottom). Finally, before PCA, we normalize the average and standard deviation of each column to make every task, regardless of the magnitude of its accuracy, have similar importance in the PCA.

D Detailed Analysis of Length Distribution Effects

This appendix elaborated on the observed effects of training data length distribution on model behavior, as summarized in the main paper.

- **Interpolation Effect in Generation Tasks:** As shown in Figure 10, fine-tuning on generation tasks (e.g., Magicoder) leads to output lengths on other generation tasks (e.g., GSM8K, Goat) that often represent a blend. The model doesn't strictly adhere to the new dataset's length profile nor entirely retain the base model's, but rather finds an intermediate distribution.

- **Classification Task Influence on Generation Length:** Fine-tuning on classification tasks (e.g., Amazon Sentiment) generally preserves the base model's generation length distribution when tested on generation tasks (Figure 11). The fine-tuning seems to focus more on discriminative features for classification rather than altering fundamental generative properties like typical output length, unless the classification dataset itself has a very strong and unusual length characteristic.

- **Dataset-Specific Tendencies:** M[Pile] - toxicity - led to significantly longer outputs on generation tasks compared to other classification datasets, indicating that dataset-specific length characteristics can be transferred as latent traits (Figure 12).

- **Length Bias in Classification:** Analysis of token lengths per class (e.g., in IMDB, PAWS,

Dataset	Domain	Characteristics	Class Labels / Distribution
MetaMath (Yu et al., 2024)	Math	Multi-step reasoning	Generation
Goat (Liu and Low, 2023)	Math	Short arithmetic	Generation (95% inputs < 500 chars)
GSM8K (Cobbe et al., 2021)	Math	Grade school math problems	Generation (75% inputs < 25 chars)
Magicoder (ISE-UIUC, 2023)	Code	Code reasoning/generation	Generation (80% input < 1.5k chars)
PAWS (Zhang et al., 2019)	NLI	Paraphrase identification	2-way (paraphrase/not), 50% each
MNLI (Williams et al., 2018)	NLI	Natural language inference	3-way (0-contradiction, 1-entailment, 2-neutral), 33.3% each
Flipkart Sentiment (KayEe)	Sentiment	Customer reviews	3-way (81.2% positive, 13.9% negative, 4.9% neutral)
Amazon Reviews (Zhang and Yassir, 2022)	Sentiment	Product reviews	5-way (1-5 stars), 20% each
IMDB Reviews (Maas et al., 2011)	Sentiment	Movie reviews	2-way (1-positive, 0-negative), 50% each
Pile (Toxicity) (Korbak, 2024)	Toxicity	Text toxicity detection	2-way (1-toxic, 0-non-toxic), 50% each

Table 6: Overview of datasets used in experiments.

Amazon - see Figure 15) reveals that base models can have biases (e.g., shorter sentences as positive sentiment). Fine-tuning can either reinforce or alter these biases depending on the training data's length characteristics per class. For example, the PAWS analysis (Figure 15) showed that the base model was biased towards shorter sentences, which Flipkart and Magicoder inherit. However, models fine-tuned on PAWS and Pile show different length biases when evaluated on IMDB (Figure 13) Similarly, IMDB analysis (see Figure 15) showed that the base model is biased towards shorter sentences as being positive.

E Length Bias for Classification Tasks

Figures 13, 14, 15 show how the base and adapter do in predicting classes in a classification task and their distribution across True Positive, False Positive, True Negative and False Negative.

F Detailed Analysis of Classification Adapter Effects on Math Performance

This appendix provides a more detailed look at how fine-tuning on classification datasets impacts performance on mathematical reasoning tasks, particularly GSM8K.

F.1 Linguistic Feature Importance Details

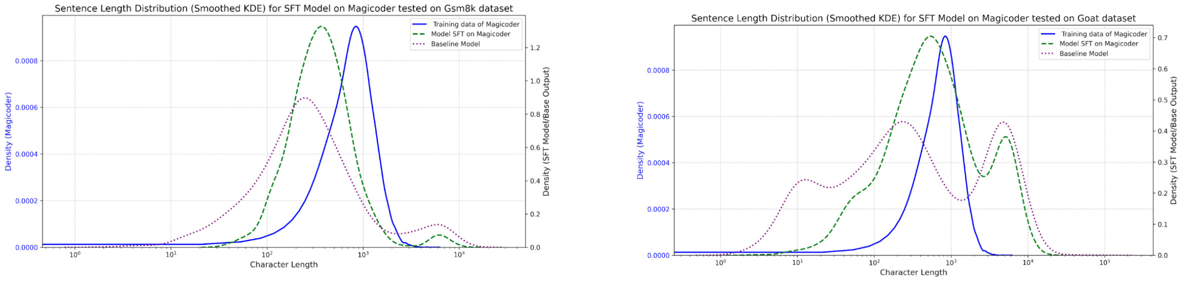
The improvement from classification adapters on GSM8K appears linked to enhanced sensitivity to linguistic structures crucial for understanding word problems.

- **Correlation of Math Features:** We analyzed the correlation between various mathematical features in word problems and the improvement in model performance when using adapters (Figure 16). Features like "num_values", "has_comparison", and "num_entities" showed negative correlations, suggesting problems with these features are less likely to show improvement with the tested adapters. Conversely, features like "has_unit_conversion" and "num_questions" showed positive (or less negative) correlations, indicating adapters might handle these better.
- **Part-of-Speech (POS) Tags:** Comparing POS tag distributions in problems solved successfully by adapter-tuned models versus the base model reveals differences (Figure 18). For instance, if adapter success cases show a higher count of *NOUNs*, it suggests adapters better handle noun-rich problems.

F.2 Analysis by Number of Steps

We defined a heuristic "number of steps" to loosely quantify problem complexity by summing counts of questions, explicit sentences, mathematical operations, comparisons, and conditional statements.

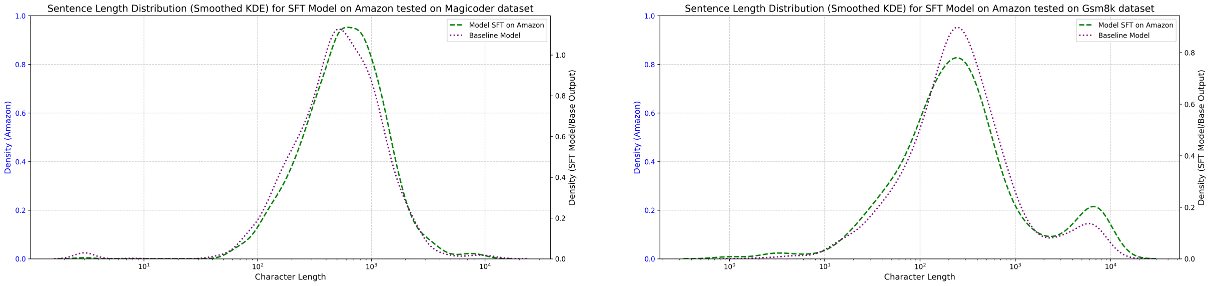
- Generally, adapter-tuned models showed varying performance improvements over the base



(a) M[Magicoder], tested on GSM8K.

(b) M[Magicoder], tested on Goat.

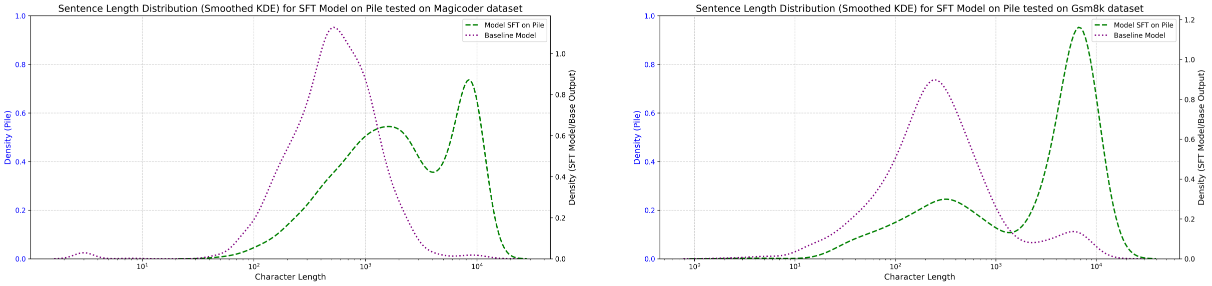
Figure 10: Generation length distribution when fine-tuned on a generation task (M[Magicoder]) and tested on other generation tasks.



(a) M[Amazon], tested on Magicoder.

(b) M[Amazon], tested on GSM8K.

Figure 11: Generation length distribution when fine-tuned on a classification task (M[Amazon]) and tested on generation tasks.



(a) M[Pile], tested on Magicoder.

(b) M[Pile], tested on GSM8K.

Figure 12: Generation length distribution when fine-tuned on Pile and tested on generation tasks.

model depending on the number of steps, often outperforming for lower to moderate step counts (Figure 19).

- A peculiar dip in adapter performance relative to the base model was consistently observed for problems estimated to have 10 steps (Figure 20). Analysis of these 10-step problems revealed they predominantly involved 'money' domain and 'multiplication' or 'addition' operations (Figures 21a, 21b). The base model excelled on these specific 10-step problems, while adapter performance decreased.

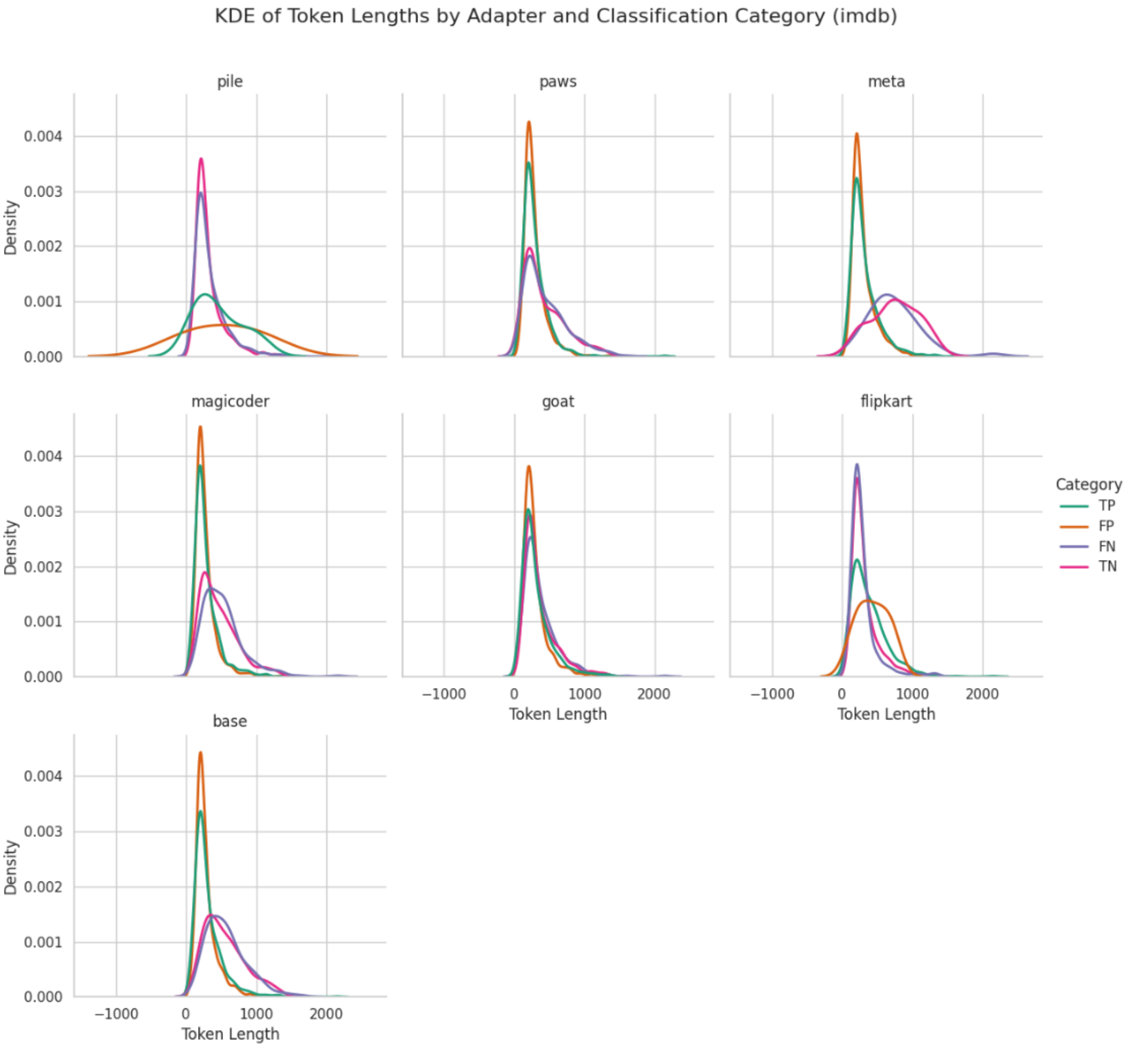


Figure 13: KDE of Token Lengths by Adapter and Classification Category (IMDB). TP: True Positive, FP: False Positive, FN: False Negative, TN: True Negative.

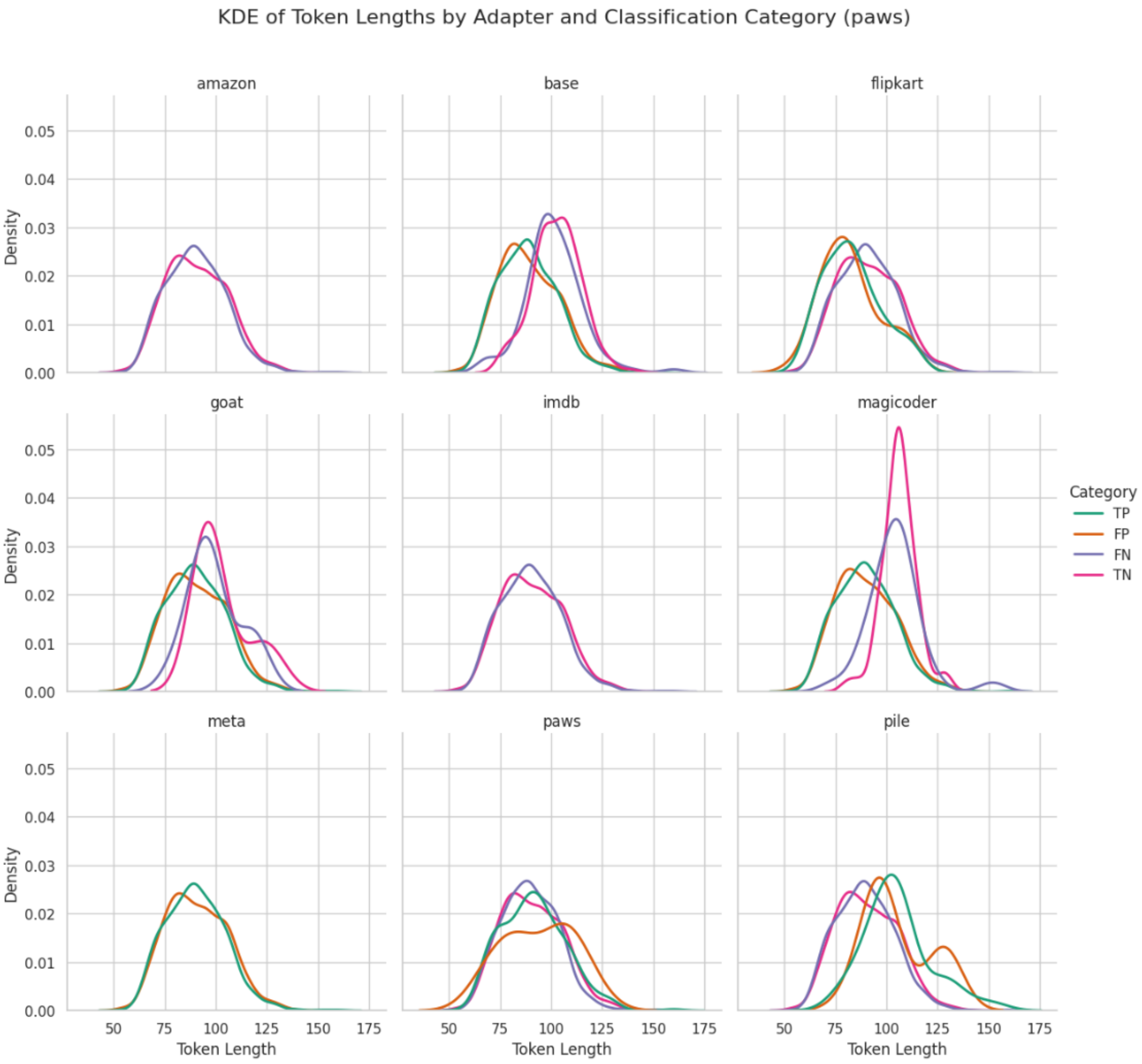
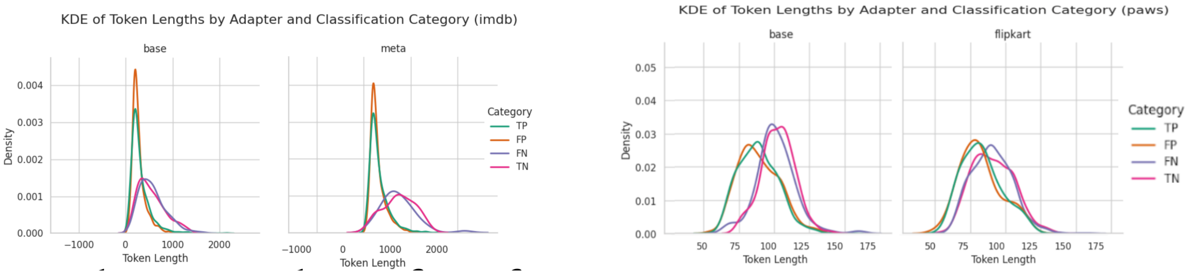


Figure 14: KDE of Token Lengths by Adapter and Classification Category (PAWS). TP: True Positive, FP: False Positive, FN: False Negative, TN: True Negative.



(a) KDE of Token Lengths of Original LLM and Meta-Math on IMDB

(b) KDE of Token Lengths of Original LLM and Flipkart on IMDB

Figure 15: KDE of Token Lengths by Adapter and Classification Category. TP: True Positive, FP: False Positive, FN: False Negative, TN: True Negative.

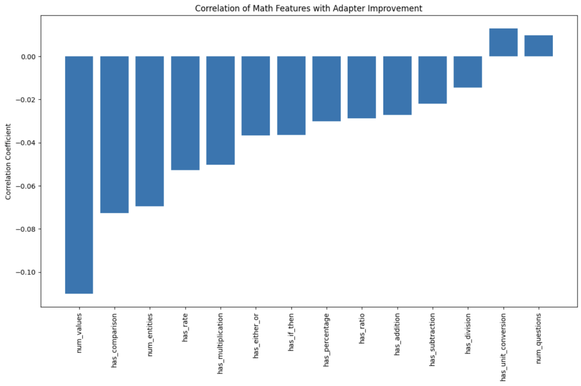


Figure 16: Correlation of Math Features in Word Problems with Adapter Improvement on GSM8K.

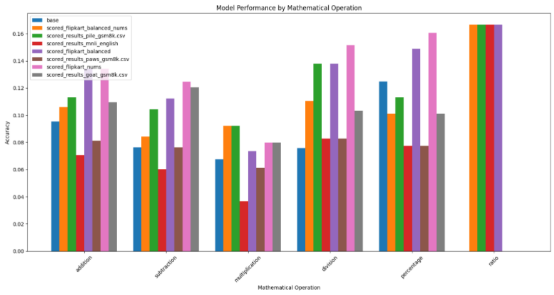


Figure 17: Adapter wise performance improvement on GSM8K clustered by arithmetic operations.

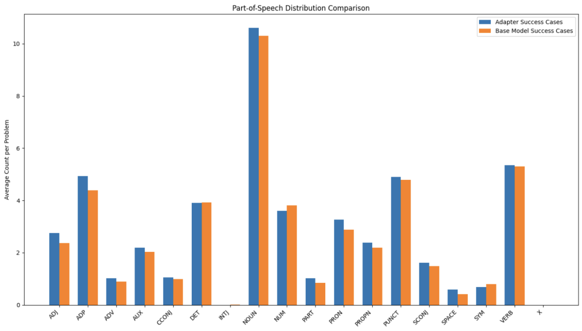


Figure 18: Part-of-Speech Distribution Comparison in GSM8K Problems (Adapter Success Cases vs. Base Model Success Cases).

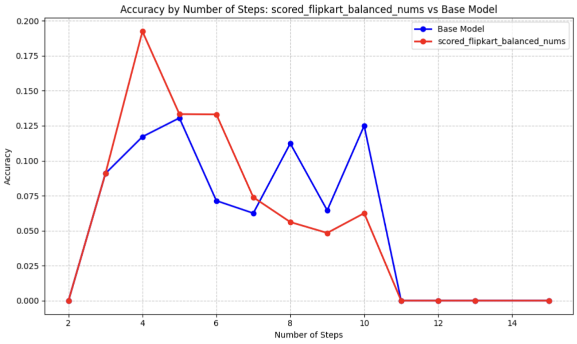


Figure 19: Accuracy by Number of Steps: M[Flipkart Balanced (Numeric)] vs Original LLM on GSM8K.

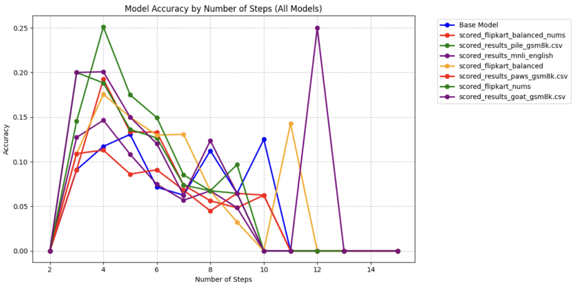
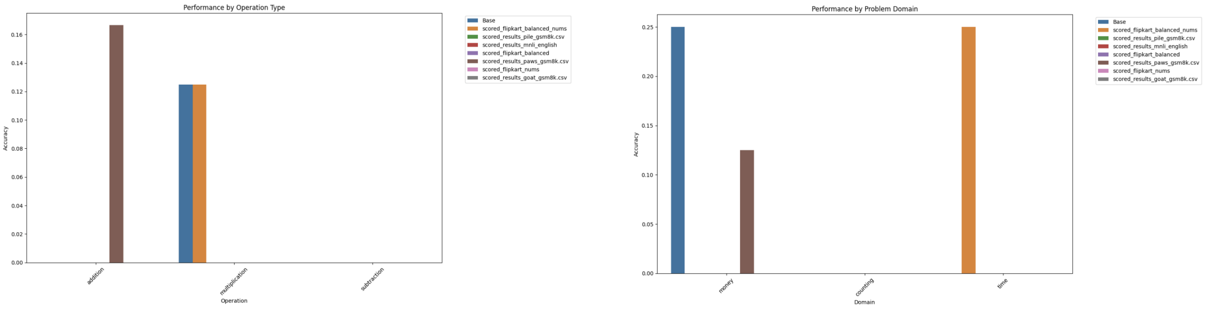


Figure 20: Model Accuracy by Number of Steps on GSM8K, highlighting the 10-step region.



(a) Performance by Operation Type for 10-Step Problems.

(b) Performance by Problem Domain for 10-Step Problems.

Figure 21: Analysis of 10-Step GSM8K problems where base model outperforms adapters.

LLMs as annotators of argumentation

Anna Lindahl
Språkbanken
University of Gothenburg
Sweden
anna.lindahl@svenska.gu.se

Abstract

Annotated data is essential for most NLP tasks, but creating it can be time-consuming and challenging. Argumentation annotation is especially complex, often resulting in moderate human agreement. While large language models (LLMs) have excelled in increasingly complex tasks, their application to argumentation annotation has been limited. This paper investigates how well GPT-4o and Claude can annotate three types of argumentation in Swedish data compared to human annotators. Using full annotation guidelines, we evaluate the models on argumentation schemes, argumentative spans, and attitude annotation. Both models perform similarly to humans across all tasks, with Claude showing better agreement with humans than GPT-4o. Agreement between models is higher than human agreement in argumentation scheme and span annotation.

1 Introduction

Annotated data is essential in most natural language processing (NLP) tasks, including semantic and pragmatic analysis. While pretrained large language models (LLMs) have reduced the need for large amounts of annotated training data, labeled data remains crucial for evaluation. However, creating high-quality annotated data can be time-consuming and expensive, especially when faced with the complex aspects of linguistic meaning involved in annotating a phenomenon like argumentation.

Annotating argumentation is a challenging task, as it involves not only identifying opinions but how they are argued for. Argumentation in itself can often be implicit and context-dependent, and sometimes even subjective, which can lead to differing opinions among annotators. In NLP, the study of argumentation is usually done within the field of argumentation mining, which aims to automatically retrieve and analyze argumentation (Stede and Schneider, 2018; Lawrence and Reed, 2020). Because of the complexity of the task, argumentation annotated datasets used in this field often report lower agreement than annotation of other phenomena in NLP (Lytos et al., 2019; Lindahl and Borin, 2024). These challenges make argumentation especially suitable for investigation.

In recent years, LLMs have excelled at different complex tasks, often outperforming previous baselines (Brown et al., 2020; Chowdhery et al., 2022). Often, these models are not fine-tuned on data, but instead instructed through prompts to perform various tasks, such as classification. This way of prompting is more similar to annotation of data, rather than training and then classifying. This similarity between prompting and annotation has given rise to several studies comparing how well LLMs annotate (Pavlovic and Poesio, 2024), with potential advantages in speed and cost (Ding et al., 2023). Recently, models are able to handle longer inputs (compare OpenAI's GPT-4's context window of 8102 tokens to GPT-4o's 128k), making it possible to prompt models with actual annotation guidelines rather than shortened instructions.

Despite the capabilities of LLMs, not many studies have yet compared argumentation annotation of humans to that of models. Because of this, in this study, three Swedish datasets annotated with different types of argumentation are used to evaluate how well two LLMs, GPT-4o and Claude Sonnet, can annotate argumentation. The datasets are annotated for argumentation schemes, spans of argumentation and attitude, respectively. Because these tasks are complex and human annotators often disagree, we are also interested to see how the models annotate in cases of disagreement. For example, is there some annotators the models align with more? More specifically, this study asks:

1. How well can LLMs annotate argumentation, given annotation guidelines?

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 242–252
November 8-9, 2025 ©2025 Association for Computational Linguistics

2. How do models annotate when humans disagree?

In answering these questions, this paper contributes a novel analysis of the models' capabilities as annotators of argumentation in three different tasks. As far as the author is aware, there are no other studies which analyse argumentation annotation of these kinds. We find that both models can annotate similarly to humans in all argumentation annotation tasks, with Claude showing better agreement with humans than GPT-4o. In the argumentation scheme and argumentation span task the agreement between the two models is higher than between human annotators.

2 Related work

In many tasks, it has been shown that LLMs can perform comparably to human annotators. For example, Gilardi et al. (2023) use ChatGPT for four annotation tasks (stance, topic, relevance, and frames) and find that the model performs similarly to, or better than, human annotators in these tasks, compared to an aggregated gold standard. Similarly, in Aldeen et al. (2023) GPT's performance in several classification tasks is presented. They find that the model overall performs well but struggles with sarcasm and emotion detection. Other areas where LLMs have been used for annotation are stance classification (Liyanage et al., 2024) and grammatical annotation (Morin and Marttinen Larsson, 2025). However, not all studies find that LLMs perform well. LLMs seem to struggle with more complex tasks, for example Wei et al. (2024) find that LLMs under-perform in the task of event extraction.

As mentioned in the previous section, the difference between annotation and classification in these kinds of studies is not always clear. In some studies (for example in Liyanage et al. (2024) models are not given instructions similar to that of what a human annotator would receive, but instead a shortened version. As pointed out by Pavlovic and Poesio (2024), most studies compare model output to a curated gold standard, without direct comparison to the other annotators.

An exception to this is Rønningstad et al. (2024), who annotate entity-level sentiment in Norwegian texts by prompting ChatGPT. They compare the model's and five human annotators' annotations to a curated dataset. Accuracy and Cohen's κ is lower for ChatGPT, with the exception of an outlier annotator. They also find that ChatGPT's errors deviate from the other annotators' labels more than the human annotators' labels. Another example is the study by Li and Conrad (2024), who annotate stance using open source LLMs. They find that LLMs show lower agreement with human annotators in cases where human annotators themselves disagree. They also find that in these cases, the stance is less explicit than in other examples.

In argumentation mining, studies have shown that LLMs can perform well on different tasks. For example, Abkenar et al. (2024) perform argument discourse unit classification and relation classification using open source models. Chen et al. (2024) explore the argumentation mining tasks of detecting claims, stance and evidence types. They find that GPT-3.5-Turbo performs best on complex tasks. Cabessa et al. (2025) fine-tune open source models and achieve state of the art results for component classification, relation classification and identification. Gorur et al. (2025) find that open source LLMs can outperform the baseline in identifying argumentative relations. There are also examples of LLMs being used to generate argumentation (Rocha et al., 2023). However, LLMs have not been successful at all tasks. Ruiz-Dolz and Lawrence (2023) find that GPT-4 performs below other models in their fallacy detection and classification task.

When looking specifically at annotation of argumentation, there are few studies. Mirzakhmedova et al. (2024) examine the annotation of argument quality. In this task, inter-annotator agreement (IAA) among human annotators is between 0.37–0.40 (expert and novice annotators). They calculate IAA across several runs with the same model, treating each run as a new annotator, and it is significantly higher (between 0.73–0.98). They also compare model annotations to human annotations in cases where there is perfect agreement between human annotators, and find that there is moderate agreement between the models and humans for most categories. Schaefer (2025) investigate how LLMs can aid in annotation of sematic argument type and find that the models can perform the task but with similar performance to a BERT model. They did not compare their results to individual human annotators.

3 Datasets

These datasets represent different annotation approaches in argumentation mining, with varying complexity, document length and genre. The datasets are all in Swedish. All datasets were annotated by annotators with a background in linguistics and with Swedish as their native language. Letters are used to represent the annotators below, for example 'annotator A'. Note, however, that the annotators differ between the annotation tasks. The annotations of these datasets are not available online, thus the risk of the model being informed by the annotations beforehand is minimal. Dataset statistics are seen in Table 1 below. Note that in the following sections all examples are translated from Swedish to English.

Annotation	Documents	Tokens	Avg. doc. size
Arg. schemes	30 editorials	20561	685
Arg. spans	9 threads	28465	3162
Attitude	500 tweets	15510	31

Table 1: Dataset statistics

3.1 Argumentation schemes

This dataset consists of 30 Swedish editorials from (Lindahl et al., 2019), with topic such as energy politics and unemployment. The editorials are annotated with Walton's argumentation schemes by two annotators. An argumentation scheme describes how inferences are being made in an argument, for example "Argument from popular opinion". The annotation task consisted of finding arguments, made up by a conclusion and one or more premises. This argument was then labeled with an argumentation scheme. Components were annotated as spans, and a span could have multiple roles (e.g. conclusion in one argument and premise in another) and be used several times. An example of an annotated argumentation scheme is given below.

Premise: 'But against this, one must weigh the obvious risks that an expansion of nuclear power entails.'
Conclusion: 'The waste must be stored for hundreds of years.'
Scheme: ARGUMENT FROM CORRELATION TO CAUSE

In this task, the annotators were instructed to use the book by Walton et al. (2008), which introduces and describes argumentation schemes, as guidelines. Although the book covers many different argumentation schemes, the number of scheme types available for annotation was restricted to 30. Because it was not feasible to provide the whole book to the model, a list of descriptions of these 30 schemes was used instead. Below is an example of a scheme description.

ARGUMENT FROM SIGN:
Premise: A is true in this situation.
Premise: Event B is generally indicated as true when its sign, A, is true in this kind of situation.
Conclusion: B is true in this situation.

Descriptions of the scheme types mentioned in this paper are found in appendix B. For a more detailed description of the annotation process, see Lindahl et al. (2019).

3.2 Argumentative spans

This dataset consists of 9 threads from two Swedish online discussion forums (Lindahl, 2020). They are annotated with argumentative spans by 8 annotators. The guidelines are approximately 2,800 tokens, with examples and diagnostic tests. The guidelines also provide a definition of argumentation and a discussion of what is to be considered argumentative. An annotated example, agreed upon by most annotators is seen below. Bold indicates argumentation.

"**I think we should eliminate home economics in schools. I consider it degrading to women.** The 1800s called and wants the school's view of women back. What do you others think?"

For a more detailed description of this annotation process and annotation disagreement, see Lindahl (2020) and Lindahl (2024).

3.3 Attitude

This dataset consists of 4280 tweets from Swedish political parties and party leaders (Lindahl, 2024, 2025). The aim of the annotation was to identify what the tweet author expressed an attitude or stance towards. This was done by marking spans of text that represented what the author expressed negative or positive attitudes about. For example, see below where bold indicates negative attitude:

> "Now every penny needs to go towards counteracting **the municipal cri-**

> **sis.** Therefore, we say no to **increased Swedish EU fees.** The EU bureaucrats will have to cut their coat according to their cloth.”

The guidelines describe the task and provide examples. This annotation study employed four annotators, and each tweet was annotated by a combination of three annotators. A subset of the tweets was annotated by all four annotators. From the tweets annotated by all four annotators, 500 were chosen for annotation in this study. For a more detailed description of the annotation process and annotation disagreement, see Lindahl (2025) and Lindahl (2024).

4 Method

For this study, the models GPT-4o (`gpt-4o-2024-08-06`) (OpenAI et al., 2024) and Claude Sonnet (`claude-3-7-sonnet-20250219`)[1] were chosen. For both models, the temperature was set to 0. This was to make the results more consistent and deterministic. GPT-4o was chosen because it is one of the most prominent and well-known models, and one of the most commonly used ones in annotation studies. It was also is one of the most cost effective models. Claude was chosen to provide a comparison with a different model architecture and it also performed well in preliminary experiments. Initially, in order to compare the results to an open source model, experiments were run with Llama 3.3-8B.[2] However, this model did not perform well enough to be included in this study, possibly due to the model size. A larger Llama model was not used due to computational constraints.

For each annotation task, the model was given a prompt which consisted of the original annotation guidelines together with short supplemental instructions. The prompt also included an example to be annotated. Both the prompt and examples were all in Swedish. The length of the guidelines is seen in Table 2.

	Editorials	Online forum	Tweets
Tokens	3426	2786	2349

Table 2: Length of the guidelines, tokens

The guidelines were not changed from the original annotation task, with the exception of removal of tool-specific instructions. The supplementary instructions introduced the task and the guidelines, and told the model in which format it should return the annotations.

As previous studies have reported (Rønningstad et al., 2024; Atreja et al., 2024), care must to be taken when crafting instructions. When writing instructions in this study, variations in wording were found to influence the annotations, and developing instructions that produced the correct output took some trial and error. In this process, both models were included, in order to develop instructions that worked for both of them. For example, mentioning the guidelines early in the instructions would increase the number of annotations returned in the correct format. An example of a prompt can be seen below, see appendix A for the other.

> “Your task is to annotate text spans that you consider to contain argumentation. Here are the annotation guidelines [guidelines]. The text is from an online forum, where each post is marked with "==". Divide the annotations per post. Make the annotations in json format, as a single object. Here is the text: [document].”

The prompt was sent for each document to be annotated. In the case of the argumentation schemes and the argumentative spans, a document would be an editorial or a thread, respectively. For the attitude annotation, ten tweets were sent at a time.

The guidelines for the argumentative spans and attitude tasks included annotated examples. In the argumentation scheme task, the guidelines did not include annotated examples. This was done in order to keep the annotation task as similar as possible to the original annotation setup.

5 Results

In general, the models struggled with keeping within the instructions and often added extra knowledge. For example, they would add a motivation for the annotation despite being explicitly asked not to. The models would also often correct the spelling of misspelled words, which made it challenging to automatically compare annotations. Some examples were returned without annotation or annotated in the wrong format. Due to these issues, around 100

---

[1] `https://docs.anthropic.com/en/docs/about-claude/models/overview`

[2] `https://www.llama.com/models/llama-3/`

tweets could not be included in the evaluation. In the editorials and the online threads, all examples were included.

5.1 Argumentation schemes

This task concerned the annotation of argumentation schemes in editorials. Comparing the annotated editorials, we found that Claude annotated more tokens than the annotators, while GPT-4o annotated fewer tokens, as seen in Table 3. GPT-4o annotated about half as many tokens as the two annotators. However, for both models the number of annotated tokens varied between editorials, sometimes matching the same number of tokens as the human annotators. Annotator A and Claude annotated a similar number of arguments, and likewise did annotator B and GPT-4o.

	A	B	GPT-4o	Claude
Anno. tokens	62%	59%	34%	71%
Arguments	345	195	187	372
Avg. nr. premises	1.26	2.03	1.15	1.33

Table 3: Annotation statistics

Looking at types of argumentation schemes, the models annotated many of the same schemes as the humans, but not necessarily for the same arguments. The most common scheme for both models were 'Argument from Consequences' (31% of the schemes for GPT-4o and 23% for Claude). This is the second most common scheme for both human annotators. The second most common scheme for GPT-4o, 'Argument from Example', was not used at all by the other annotators. The second most common scheme for Claude, 'Argument from cause to effect', which is the fourth most common scheme for both annotators.

In this annotation, an annotated span representing a component can be used in more than one argument, and take on both the role of premise and conclusion (in separate arguments). For this reason, agreement is first compared on token-level and then on individual arguments. Table 4 shows agreement in Krippendorff's α (Krippendorff, 2018) between annotators and models on whether a token is annotated as being part of an argument component, as well as agreement on the separate component types. Overall, Claude agrees more with the human annotators than GPT-4o does. Adding Claude's annotation to the humans annotations increases the agreement. GPT-4o agrees more on conclusions than on premises, while the agreement between Claude an humans is similar for both components. Interestingly, the highest agreement is between the two models, without including the human annotators. It should be noted that agreement in general is quite low, reflecting the complexity of the task.

	Arg. vs. non-arg	Conclusion	Premise
H	0.15	0.19	0.22
GPT+H	0.03	0.12	0.074
CL+H	0.16	0.16	0.18
CL+GPT	0.26	0.23	0.21
All	0.13	0.15	0.13

Table 4: α for the schemes. H = Humans, GPT = GPT-4o, CL = Claude

Agreement between humans and models varied between editorials, between 0.26– -0.26 for GPT-4o and between 0.38– -0.05 for Claude. One editorial had lowest agreement between both models and humans, as well as low agreement between the human annotators. Inspecting this editorial the annotators (both models and humans) have found different arguments. However, as seen in the example below, one conclusion was annotated by all, but together with different premises and schemes.

Conclusion All: 'It is not difficult to understand that the Social Democrats talk more about how many people are employed than about how many lack employment.'

Premise A: 'If there are many who are unemployed, it is bad to talk about how many lack employment.'

Premise A: 'Many are unemployed'

Premise B: 'The Social Democrats have, as is well known, replaced their old slogan "full employment´´ with 'work for all."

Premise B: 'When it comes to 'work for all,' the Social Democrats are more vague about the goal.'

Premise Claude: 'When it comes to 'work for all,' the Social Democrats are more vague about the goal.'

Premise GPT-4o: 'At his press conference in Malmö on Friday, the Prime Minister, apparently deliberately, downplayed the significance.'

Figure 2 shows pair-wise agreement between annotators. Both models agree more with annotator A, and Claude agrees the most with all.

Looking instead at the individual arguments and their components, there are 25 arguments between all with the same conclusion. Out of these 25, 6 share at least one premise. In these arguments,

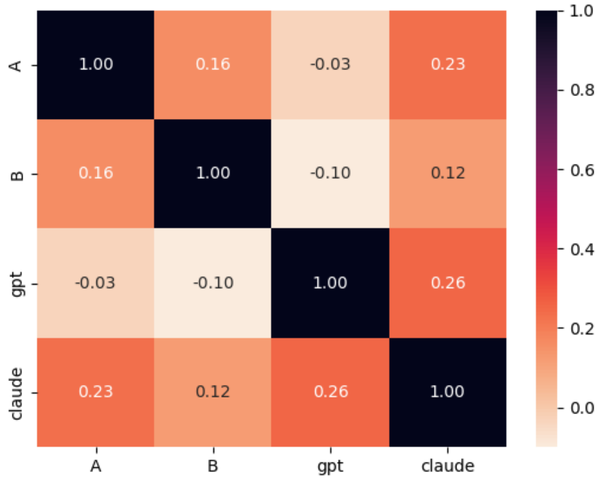


Figure 1: Pair-wise α on token-level, for the schemes

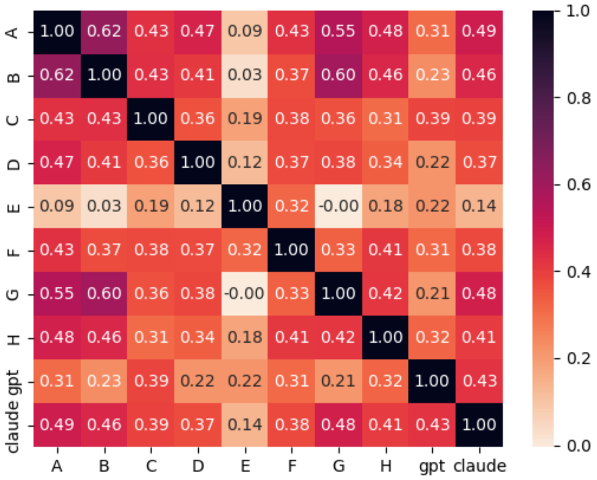


Figure 2: Pair-wise α for the online forum

the models often agreed on the scheme, while the humans more often disagreed.

Manually inspecting the annotated arguments further, more examples of annotators (both models and humans) choosing the same conclusion but different schemes and premises is found. Likewise, there are examples of components being used as both premise and conclusion, as in the example below.

Premise GPT-4o& A, Conclusion Claude: 'On election day, the individual voter is sovereign.'

Premise Claude & A, Conclusion GPT: 'This is the foundation of democracy.'

Conclusion A: 'Therefore, our appeal to our readers is this: take the opportunity to decide how our country should be governed over the next three-year period.'

Scheme A: ARGUMENT FROM CONSEQUENCES

Scheme GPT-4o: ARGUMENT FROM POPULAR OPINION

Scheme Claude: ARGUMENT FROM POSITION TO KNOW

5.2 Argumentation spans

In this task, the models were asked to annotate threads from online forums with spans that they considered to be argumentation. Similar to the previous task, GPT-4o annotated fewer tokens than most of the other annotators in the online forums. The model annotated 17.4% of the tokens as argumentative, as compared to between 20–44% for the other annotators. There is however an outlier annotator who has annotated even fewer tokens. Claude annotated 33.6% as argumentative. Agreement on token level is seen in Table 5. Agreement between the two models and humans, both together and separately is very close to the agreement between humans (0.39). However, agreement between only the models is slightly higher (0.43).

	H	GPT+H	CL+H	GPT+CL	All
α	0.39	0.37	0.39	0.43	0.38

Table 5: α on token level, argumentation spans. H = Humans, GPT = GPT-4o, CL = Claude

In Figure 2, pairwise agreement is shown. GPT-4o had among the lowest inter-annotator agreement scores, while Claude achieved higher agreement with human annotators. GPT-4o agrees the most with annotator C and H and Claude agrees the most with A and G. Interestingly, GPT-4o agrees more with the outlier annotator E than most of the other annotators do.

The pairwise token overlap between the annotators was also compared, as seen in Figure 3. For 4 of the annotators (A,B,G and F) more than 50% of GPT-4o's and Claude's annotations overlap with their annotations. In manual inspection of the threads, examples of overlap and partial overlap were also found. In the examples with high agreement, it was found that the human annotators and Claude often annotated longer spans, while GPT-4o did not. For example, below all annotators (including the models) annotated the first part of the text as argumentation, and while Claude and four of the human annotators also annotated the part in italics.

> "On the contrary. We need more home economics in schools, and more subjects need to be integrated into home economics. Home economics is the subject

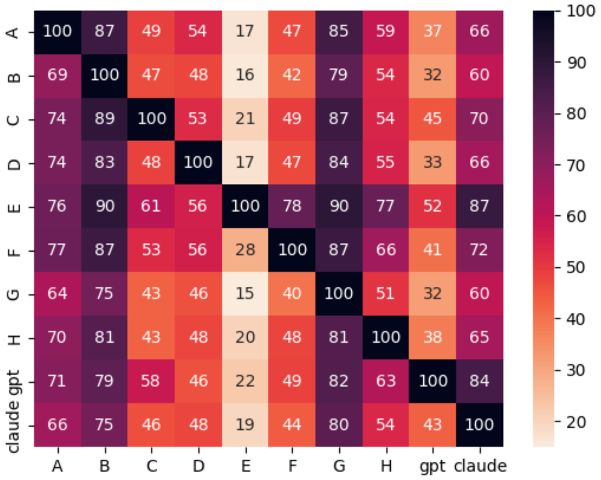


Figure 3: Pairwise percentage overlap. The figure is not symmetrical, for example are 71% GPT-4o's annotations found in A's, but 37% of A's found in GPT-4o's.

> that truly has the potential to teach practical, real-world knowledge that young people need in order to manage on their own. *And don't come and say that children learn this kind of thing at home anyway, because that's actually far from certain.*"

When inspecting spans with low agreement, the models sometimes annotated spans that were probably not intended as argumentation. For example, the span below was only annotated by one of the human annotators and is more narrative than argumentative, but the models both annotated it as argumentative.

> "After struggling for years to improve the situation without success, I have decided to leave."

5.3 Attitude

In this task, tweets were annotated for object of negative or positive attitude. The number of tweets to be annotated was originally 500, but 97 of them were excluded either due to wrong annotation format or missing annotations. As seen in Table 6, out of the remaining tweets, GPT-4o annotated the fewest tokens (10%), while Claude annotated similarly to the human annotators. Comparing label distribution, the models annotated a similar amount of negative and positive tags, while the human annotators annotated more positive tokens than negative ones.

	A	B	C	D	GPT	CL
POS	0.67	0.63	0.71	0.62	0.52	0.56
NEG	0.33	0.37	0.29	0.38	0.47	0.43
Tok.	52%	23%	25%	25%	10%	33%

Table 6: Label distributions in annotations. Tok. = annotated tokens. GPT = GPT-4o, CL = Claude

Agreement on token level is shown in Table 7. Unlike the previous tasks, agreement between the two models is lower than other annotator combinations. Agreement within humans and humans + Claude is similar, while GPT-4o has lower agreement with humans.

	H	GPT+H	CL+H	GPT+CL	All
α	0.35	0.27	0.35	0.25	0.30

Table 7: α on token level, argumentation spans. H = Humans, GPT = GPT-4o, CL = Claude

Pair-wise agreement is shown in Figure 4. It is lower between GPT-4o and the other annotators, while Claude's agreement scores are more similar to the human annotators.

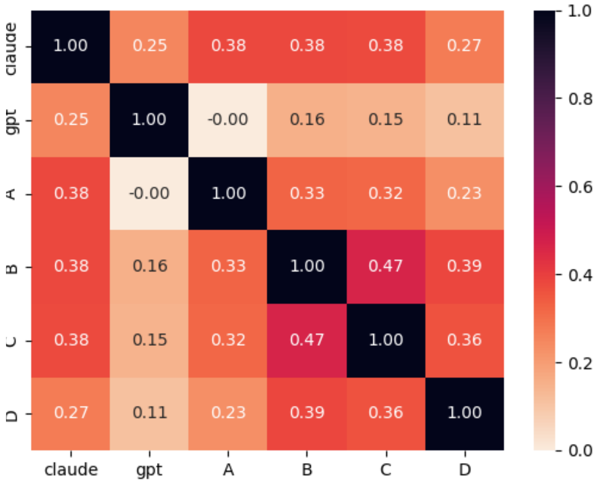


Figure 4: Pair-wise α for the tweets

For investigating labels further, Krippendorff's unitized $_u\alpha$ is employed (Krippendorff et al., 2016). This measure is suitable for comparing spans, but it can also show agreement on only labeled spans (ignoring label combinations such as NEG,O). As seen in Table 8, agreement is high for all annotator combinations. This tells us that the annotators and models agree substantially on the label, in instances when they have annotated the same span.

Annotators	$_{posu}\alpha$	cover	$_{negu}\alpha$	cover
H	0.82	53%	0.82	53%
GPT+H	0.83	44%	0.83	45%
CL+H	0.83	53%	0.83	54%
GPT+CL	0.97	36%	0.97	39%
All	0.84	47%	0.84	48%

Table 8: $_u\alpha$ for labels. Cover = coverage, % of annotated spans. H = Humans, GPT = GPT-4o, CL = Claude

Manual inspection of examples with low and high agreement revealed that low agreement often resulted from one annotator considering something argumentative that the others did not. However, it was rarely the case that the models annotated something which had not been annotated by at least one other annotator. In cases of disagreement between annotators, there was no annotator who the models with aligned more. In general, there were examples both where the models seemed to be better at following the guidelines, and cases where they annotated strangely. The models also often annotated shorter spans than the human annotators.

An example of this is shown below, where **bold** signifies positive spans, *italics* negative. In the example, “aid is needed” is not an obvious object. On the other hand, in the first sentence the models have annotated “Humanitarian superpower”, which is more in line with the guidelines which instructed the annotators to keep the spans as short as possible. The other annotators have instead annotated the full sentence.

- GPT-4o: Sweden should continue to be a **humanitarian superpower.** Our **aid is needed. Humanitarian organizations** are *shamefully underfunded.*

- Claude: Sweden should continue to be a **humanitarian superpower. Our aid is needed. Humanitarian organizations** are *shamefully underfunded.*

- A & C: **Sweden should continue to be a humanitarian superpower. Our aid** is needed.*Humanitarian organizations are shamefully underfunded.*

- B: Sweden should continue to be a humanitarian superpower. **Our aid** is needed. Humanitarian organizations are shamefully underfunded.

- D: **Sweden should continue to be a humanitarian superpower.** Our aid is needed. Humanitarian organizations are shamefully underfunded.

What is included in a span can also affect the label, as seen in the example below.

- A B, & D: Sweden now has **a government that will not introduce market rents.** Tenants are today’s big winners.

- GPT-4o, Claude & C: Sweden now has a government that will not introduce *market rents.* Tenants are today’s big winners.

6 Conclusions

This paper first examined how well the models could perform annotation tasks when provided with guidelines. In this study we have shown that both models exhibit similar annotation patterns and agreement to that of humans, which leads us to conclude that the models can follow the guidelines and perform the task reasonably well. As these are tasks where humans often disagree, reaching comparable levels of agreement to humans could be a sign that the models, especially Claude, ’understands’[3] the task.

In all tasks, GPT-4o annotated fewer tokens than the human annotators, while Claude annotated a similar amount of tokens. This might be because GPT-4o only annotates when highly confident or due to differences in how the models approach the task. While models might respond differently to prompts, GPT-4o consistently annotated fewer tokens even during the prompt design phase.

Comparing agreement with the annotators, GPT-4o agrees less with the annotators than Claude. Claude exhibits agreement similar to that of humans, as well as higher agreement in the argumentation scheme task. When manually inspecting annotated examples, Claude would often annotate more similarly to humans, while GPT-4o shows similar patterns as another, slightly worse, human annotator. However, both models’ annotations were often valid. As there can be cases where multiple interpretations are correct, for example in choosing a component as a premise or conclusion, one can not always conclude that models are wrong even if they choose to annotate differently than humans.

---

[3] The discussion of what understanding in this context means is left for another study.

However, there were some cases where the models' annotations did not make sense.

Because of the complexity of these tasks, and the fact that human annotators often disagree in them, the second question asked how models annotate when humans disagree. There were instances where models aligned with specific annotators. In the argumentation scheme task, both models agreed more with one of the annotators. Likewise, in the argumentation spans and attitude task, there was higher agreement with some annotators. However, in these tasks, the agreement is also different within the human annotators themselves. Most prominently, however, agreement was higher between the models than the humans in the argumentation scheme and spans task, possibly suggesting similarities in their reasoning. In the former task the models also agreed more on argumentation schemes types.

For future research, there are several promising directions. First, evaluating how other LLMs, particularly open-source models, annotate these datasets would help determine whether these findings generalize across different models and versions. Second, testing these models in zero-shot settings would reveal the extent of their inherent knowledge about argumentation without explicit guidelines. Finally, expanding the analysis to include other argumentation datasets, especially those in English, would provide broader insights into model performance across diverse argumentative contexts. In general, there are many questions to answer regarding how to use LLMs as annotators. For example, should each separate run be treated as a new annotator? In that case, should a failed run be considered as an annotator failing to perform the task?

Limitations

This study explores how two versions of the GPT and Claude models annotate, but the results might not hold for updated version versions of these models. Likewise, new models and new versions of existing models are released with increasing speed. This leads to difficulties reproducing results, not only in this study but for most studies employing LLMs.

The experiments in this study were carried out in the Swedish language, with instructions, guidelines and datasets in Swedish. This might limit the cross-linguistic generalizability of the results.

References

Mohammad Yeghaneh Abkenar, Weixing Wang, Hendrik Graupner, and Manfred Stede. 2024. Assessing open-source large language models on argumentation mining subtasks. *Preprint*, arXiv:2411.05639.

Mohammed Aldeen, Joshua Luo, Ashley Lian, Venus Zheng, Allen Hong, Preethika Yetukuri, and Long Cheng. 2023. Chatgpt vs. human annotators: A comprehensive analysis of chatgpt for text annotation. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 602–609. IEEE.

Shubham Atreja, Joshua Ashkinaze, Lingyao Li, Julia Mendelsohn, and Libby Hemphill. 2024. Prompt design matters for computational social science tasks but in unpredictable ways. *Preprint*, arXiv:2406.11980.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2025. Argument mining with fine-tuned large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635.

Guizhen Chen, Liying Cheng, Anh Tuan Luu, and Lidong Bing. 2024. Exploring the potential of large language models in computational argumentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2309–2330, Bangkok, Thailand. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. Can large language models perform relation-based argument mining? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8518–8534, Abu Dhabi, UAE. Association for Computational Linguistics.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity*, 50:2347–2364.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Mao Li and Frederick Conrad. 2024. Advancing annotation of stance in social media posts: A comparative analysis of large language models and crowd sourcing. *Preprint*, arXiv:2406.07483.

Anna Lindahl. 2020. Annotating argumentation in Swedish social media. In *Proceedings of the 7th Workshop on Argument Mining*, pages 100–105, Online. ACL.

Anna Lindahl. 2024. Disagreement in argumentation annotation. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 56–66, Torino, Italia. ELRA and ICCL.

Anna Lindahl. 2025. Annotating attitude in Swedish political tweets. In *Proceedings of the Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2025)*, pages 106–110, Tallinn, Estonia. University of Tartu Library, Estonia.

Anna Lindahl and Lars Borin. 2024. Annotation for computational argumentation analysis: Issues and perspectives. *Language and Linguistics Compass*, 18(1):e12505.

Anna Lindahl, Lars Borin, and Jacobo Rouces. 2019. Towards assessing argumentation annotation - a first step. In *Proceedings of the 6th Workshop on Argument Mining*, pages 177–186, Florence. ACL.

Chandreen R Liyanage, Ravi Gokani, and Vijay Mago. 2024. Gpt-4 as an x data annotator: Unraveling its performance on a stance classification task. *PloS one*, 19(8):e0307741.

Anastasios Lytos, Thomas Lagkas, Panagiotis Sarigiannidis, and Kalina Bontcheva. 2019. The evolution of argumentation mining: From models to social media and emerging tools. *Information Processing & Management*, 56(6):102055.

Nailia Mirzakhmedova, Marcel Gohsen, Chia Hao Chang, and Benno Stein. 2024. Are large language models reliable argument quality annotators? In *Conference on Advances in Robust Argumentation Machines*, pages 129–146. Springer.

Cameron Morin and Matti Marttinen Larsson. 2025. Large corpora and large language models: a replicable method for automating grammatical annotation. *Linguistics Vanguard*, (0).

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 100–110, Torino, Italia. ELRA and ICCL.

Victor Hugo Nascimento Rocha, Igor Cataneo Silveira, Paulo Pirozelli, Denis Deratani Mauá, and Fabio Gagliardi Cozman. 2023. Assessing good, bad and ugly arguments generated by chatgpt: a new dataset, its methodology and associated tasks. In *EPIA Conference on Artificial Intelligence*, pages 428–440. Springer.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. 2024. A GPT among annotators: LLM-based entity-level sentiment annotation. In *Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 133–139, St. Julians, Malta. Association for Computational Linguistics.

Ramon Ruiz-Dolz and John Lawrence. 2023. Detecting argumentative fallacies in the wild: Problems and limitations of large language models. In *Proceedings of the 10th Workshop on Argument Mining*, pages 1–10, Singapore. Association for Computational Linguistics.

Robin Schaefer. 2025. On integrating LLMs into an argument annotation workflow. In *Proceedings of the 12th Argument mining Workshop*, pages 87–99, Vienna, Austria. Association for Computational Linguistics.

Manfred Stede and Jodi Schneider. 2018. *Argumentation Mining*. Morgan & Claypool, San Rafael.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press, Cambridge.

Kangda Wei, Aayush Gautam, and Ruihong Huang. 2024. Are LLMs good annotators for discourse-level event relation extraction? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1–19, Miami, Florida, USA. Association for Computational Linguistics.

A Prompts

Translated prompts for the different tasks.

A.1 Argumentation schemes

“Your task is to annotate Walton’s argumentation schemes. Here is a description of these schemes: [guidelines]
Based on these descriptions, I want you to annotate schemes. Do this by marking out what is the conclusion and which premises belong to it. Then you mark which argumentation scheme is used. I want you to mark out exact text spans. Mark out all schemes you can find. Be thorough and don’t stop until you can’t find more schemes. Return the annotations as a single json file, in this format: {"scheme": "annotated scheme name", "components":[{"role":"conclusion","span": "the annotated conclusion"},{"role": "premise", "span": "the annotated premise"}]} Include the entire text span in the "span" field, don’t abbreviate and don’t correct any spelling errors. Annotate this text: [example] ”

A.2 Argumentation spans

“Your task is to annotate text spans that you consider to contain argumentation. Here are the annotation guidelines [guidelines]. The text is from an online forum, where each post is marked with "==". Divide the annotations per post. Make the annotations in json format, as a single object. Here is the text: [example]. ”

A.3 Attitude

“Your task is to annotate tweets. Here are the annotation guidelines . Make the annotations by marking which words are included in positive or negative spans in json format like this: [{"tweet id": 0 , "annotated tweet": Now_O needs_O every_O penny_O needs_O to_O go_O to_O counteract_O the_NEG municipal_NEG crisis_NEG ._O Therefore_O we_O say_O no_O to_O increased_NEG Swedish_NEG EU-fee_NEG ._O },{"tweet": ...}] Make sure it is valid json. Be careful to annotate what the attitude is expressed towards, not generally negative or positive words. Remember that both words, phrases and whole sentences can be annotated. Annotate these tweets: [10 examples] ”

B Scheme descriptions

ARGUMENT FROM SIGN:
Premise: A is true in this situation.
Premise: Event B is generally indicated as true when its sign, A, is true in this kind of situation.
Conclusion: B is true in this situation.

ARGUMENT FROM CONSEQUENCES:
Premise: If A is brought about, then good (bad) consequences will (may plausibly) occur.
Conclusion: A should (not) be brought about.

ARGUMENT FROM EVIDENCE TO A HYPOTHESIS:
Premise: If hypothesis A is true, then a proposition B, reporting an event, will be observed to be true.
Premise: B has been observed to be true in a given instance .
Conclusion: A is true.

ARGUMENT FROM CORRELATION TO CAUSE:
Premise: There is a positive correlation between A and B.
Conclusion: A causes B.

ARGUMENT FROM POPULAR PRACTICE:
Premise: If a large majority (everyone, nearly everyone, etc.) does A, or acts as though A is the right (or an acceptable) thing to do, then A is a prudent course of action.
Premise: A large majority acts as though A is the right thing to do.
Conclusion: A is a prudent course of action.

ARGUMENT FROM EXAMPLE:
Premise: In this case, the individual a has property F and also property G
Premise: a is typical of things that have F and may or may not have G
Conclusion: Generally, if x has property F then (usually, probably, typically) x also has property G

ARGUMENT FROM POSITION TO KNOW:
Premise: a is in a position to know whether A is true (false)
Premise: a asserts that A is true (false)
Conclusion: A is true (false)

If We May De-Presuppose: Robustly Verifying Claims through Presupposition-Free Question Decomposition

Shubhashis Roy Dipta, Francis Ferraro
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250 USA
`{sroydip1,ferraro}@umbc.edu`

Abstract

Prior work has shown that presupposition in generated questions can introduce unverified assumptions, leading to inconsistencies in claim verification. Additionally, prompt sensitivity remains a significant challenge for large language models (LLMs), resulting in performance variance as high as **3–6%**. While recent advancements have reduced this gap, our study demonstrates that prompt sensitivity remains a persistent issue. To address this, we propose a structured and robust claim verification framework that reasons through presupposition-free, decomposed questions. Extensive experiments across multiple prompts, datasets, and LLMs reveal that even state-of-the-art models remain susceptible to prompt variance and presupposition. Our method consistently mitigates these issues, achieving up to a **2–5%** improvement.[1]

1 Introduction

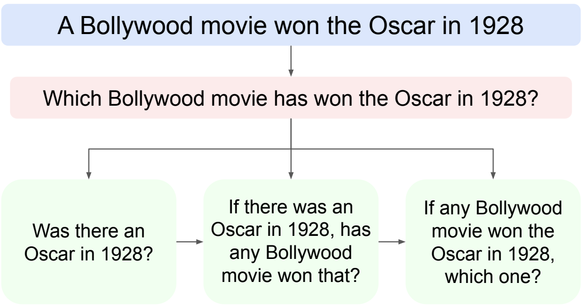

A Bollywood movie won the Oscar in 1928
Which Bollywood movie has won the Oscar in 1928?
Was there an Oscar in 1928?
If there was an Oscar in 1928, has any Bollywood movie won that?
If any Bollywood movie won the Oscar in 1928, which one?

Figure 1: In contrast to simple claim decomposition, which can presuppose some parts of the claim as true (i.e. there were Oscars in 1928), our method automatically generates a collection of presupposition-free questions to verify each part of the claims. Here, it first asks whether there were Oscars in 1928, conditioning any subsequent questions on that answer.

While current large language models (LLMs) (Dubey et al., 2024; Touvron et al., 2023; DeepSeek-AI et al., 2025; Qwen et al., 2025) demonstrate strong performance in claim verification (Tang et al., 2024; Kamoi et al., 2023) when provided with ground truth evidence, they can improperly presuppose parts of the claim, which can then lead to incorrect conclusions or explanations. Consider, for example, the claim illustrated in Fig. 1: "*A Bollywood movie won the Oscar in 1928*": an LLM might simply verify whether "*any Bollywood movie won the 1928 Oscar*," thereby accepting the existence of the 1928 Oscar without verifying it first.[2] In contrast, a more skeptical verifier would first ask, "*Was there an Oscar in 1928?*" – and only if that is true, proceed to verify the rest of the claim. This multi-layered reasoning highlights the need for decomposition-based verification that explicitly questions and validates every presupposition before drawing a conclusion.

We propose such a method of question-based decomposition and verification. Specifically, **we demonstrate how presupposition-free questions can reduce prompt sensitivity, improve verification performance, and constrain overthinking**. This simple yet effective method creates structured reasoning paths that reduces prompt sensitivity by **2–5%**, offering a significant robustness gain. Our approach improves performance across both general and domain-specific scientific claim verification while reducing the need for labor-intensive prompt-tuning across datasets.

To summarize, our contributions are as follows:

1. We propose a decomposition and de-presupposition based question generation method to produce fine-grained questions.
2. We show that while LLMs are highly sensitive to prompt variations, our structured reasoning approach significantly reduces this sensitivity, improving robustness across prompts.
3. We demonstrate that our automated question

---

[1] `https://github.com/dipta007/De-Presuppose`

[2] The Oscars were first awarded in 1929.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 253–266
November 8-9, 2025 ©2025 Association for Computational Linguistics

generation achieves approximately 89% coverage of atomic subclaims in a zero-shot setting.

4. We show that our method outperforms strong baselines by 2–5% across two datasets, three prompt variants, and three reasoning models.

Our code is available at `https://github.com/dipta007/De-Presuppose`.

2 Method

Given a claim ("*Bollywood movie has won the Oscar in 1928*"), our approach proceeds in three steps: First, we automatically **decompose the claim into simpler questions** (§2.1) – "*Which Bollywood movie has won the Oscar in 1928?*". We recognize that the generated questions may presuppose certain facts, such as "*There was an Oscar award in 1928.*" To account for this, we next **reformulate and expand these questions to remove presuppositions**, resulting in presupposition-free questions (§2.2) – "(a) *Was there an Oscar in 1928?*, (b) *If there was, has any Bollywood movie won that?* (c) *If yes, which one was that?*" Finally, we use a reasoner to **verify the claim (§2.3) with the help of these questions**. Our results show that this structuring improves claim verification performance.

2.1 Question Generation

Chen et al. (2022) demonstrated that using evidence during question generation yields significantly better results. Motivated by this, we incorporate both claim and evidence in our question generation process. Inductively, we prompt an LLM to decompose a claim into separate, independent questions[3]. We adopted the prompt from Kamoi et al. (2023) to better align with our question decomposition module instead of their subclaim decomposition. Specifically, their few-shot prompt includes examples of claim-to-subclaim decomposition, which we adapted by converting the assertive statements into questions.

2.2 Question De-Presupposition

Through manual review of the generated questions, we identified that some of them presuppose information that may or may not be true, i.e., "*Which Bollywood movie has won the 1928 Oscar?*". While a model could, in theory, decompose the claim into presupposition-free questions, in practice we noticed this did not consistently happen. Rather, we noticed that models' subsequent reasoning over those questions would go a different way than intended. To address this, we employ LLM-based prompting to process the questions and decompose any presuppositions into multiple subquestions, as illustrated in Fig. 1. Specifically, for each question, we prompt the LLM to decompose it into presupposition-free atomic question[4].

2.3 Reasoning through Questions

Next, we employ a reasoning model that utilizes both the questions and the evidence to verify the claim. The key in our approach is that the presupposition-free questions from the previous step can guide the LLM to reason effectively, instead of directly verifying the claim based solely on the evidence. We do this reasoning by prompting an LLM; however, Sclar et al. (2024) found that even large models are susceptible to prompt variance. Similarly, we found in our experiments that there was significant variation in performance based on which exact prompt we used: one was adopted from Tang et al. (2024)'s verification prompt ("MiniCheck"), and two others were written by this paper's authors in an attempt to provide more **S**tructured **G**uidance (we call them "SG1" and "SG2"). We provide full details and results in appendix (App. A.3). Overall, we found that just by using these prompts alone performance varied by up to 6% accuracy. This shows the need to mitigate prompt sensitivity; as our results will show, our method provides this ability.

3 Experiments

Evaluation Metric: We used balanced accuracy (BAcc) as the evaluation metric (Tang et al., 2024; Kamoi et al., 2023), due to the imbalance between all the labels. To evaluate our question module, we have proposed a question coverage metric to assess the accuracy of generated questions (§3.2).

Datasets: We used the BioNLI (Bastan et al., 2022) and WiCE (Kamoi et al., 2023) dataset. While BioNLI features highly complex and domain specific scientific claims, WiCE has the real-life claims from Wikipedia. We have also curated a random balanced subset (300 claim-evidence pairs) of the BioNLI dataset to experiment with much costlier models, i.e., `o4-mini`. We call it BioNLI-300. Following previous works (Tang et al., 2024;

---

[3] Prompt is provided on App. A.1.1

[4] Prompt is provided on App. A.1.2

Prompt	Only-Reasoner	Reasoner + Question Decomposition	Our Method
BioNLI-FULL			
SG2	73.74 ± 0.10	**76.57 ± 0.08**	**76.57 ± 0.19**
SG1	72.34 ± 0.26	76.72 ± 0.14	**77.73 ± 0.06**
MiniCheck	77.58 ± 0.14	78.04 ± 0.14	**78.32 ± 0.15**
BioNLI-300			
SG2	69.11 ± 0.42	72.56 ± 1.50	**73.44 ± 1.40**
SG1	68.44 ± 0.68	74.11 ± 1.29	**75.00 ± 2.18**
MiniCheck	73.33 ± 0.98	74.33 ± 1.25	**75.11 ± 1.10**
WiCE			
SG2	76.36 ± 0.27	**79.32 ± 0.49**	79.03 ± 0.00
SG1	73.41 ± 0.56	**78.23 ± 0.64**	76.42 ± 0.92
MiniCheck	80.70 ± 0.30	81.72 ± 0.50	**82.25 ± 0.62**

Table 1: Results (with standard deviation) on the BioNLI and WiCE dataset. Full results are reported in the appendix (Table 4).

Kamoi et al., 2023), we converted the WiCE dataset from a three-class problem to binary classification by considering both ‘Refuted’ and ‘Partially Supported’ as the ‘Refuted.’ Detailed dataset statistics are in the appendix (Table 8).

Experimental Setup: We have used `Qwen/QwQ-32B` and `o4-mini` as the reasoner models. Both models were run with the default temperature settings, while for `o4-mini`, we utilized the “high” reasoning setup. More details on the implementation is provided on App. A.2.

3.1 RQ1: Does de-presupposition help?

We consider three setups: (1) Only Reasoner, (2) Reasoner + Question Decomposition and (3) Our method with de-presupposition. The results of all the settings are reported in Table 1. The findings indicate that de-presupposition consistently improves performance across all settings. The full results in Table 4 also show that most notable gains are observed when paired with a more capable reasoning model, such as `o4-mini`. We hypothesize that less capable models may not fully leverage the benefits of presupposition-free questions. The impact of de-presupposition is more pronounced on complex, multi-hop datasets like BioNLI, while the improvements are less substantial on simpler datasets such as WiCE.

We also experimented with adding an explicit answer module to further ease the burden on the verifier. However, as shown in Table 5, this addition often degrades performance. We hypothesize that generating intermediate answers may introduce errors that propagate to the verifier. While a similar risk exists for question decomposition and de-presupposition, breaking a claim into sub-questions—and refining them into presupposition-free forms—is comparatively easier and less error-prone than answering questions directly from long, unstructured documents.

Question Decomposer	Question Coverage
o4-mini	89.16 ± 0.20
Qwen/QwQ-32B	87.41 ± 0.00

Table 2: Coverage of the sub-claims from the WiCE dataset. Standard deviation is across 3 runs for question decomposing.

3.2 RQ2: Do we cover sub-claims?

To analyze how well our generated questions cover the critical parts of the claim, we use the decomposed subclaims from the WiCE dataset. We employed `Qwen/Qwen3-32B` to evaluate coverage. During evaluation, for each subclaim, we asked the model whether it was addressed by at least one question or a combination of multiple questions. This approach accounts for the fact that a single subclaim may be implicitly addressed by multiple questions, aligning with our goal of generating as many skeptical and granular questions as possible. For example, the claim “San Jose is the biggest city in Texas” can be decomposed into two questions: (1) *Is San Jose a city in Texas?* and (2) *If yes, is it the biggest?* To avoid occasional contextualization issues when aligning questions with subclaims, we provided the ground truth evidence as additional context, allowing the LLM to properly de-contextualize the questions.[5]

Table 2 shows we achieve ∼89% coverage of subclaims, regardless of the underlying LLM. These results underscore that question decomposition is highly effective in zero-shot settings, while the low standard deviation across runs highlights the robustness of the question generation.

3.3 RQ3: Can we mitigate prompt variance?

In the next RQ, we explore if our proposed method can mitigate the prompt sensitivity. To analyze this question, we used our method across two datasets – WiCE (Kamoi et al., 2023) and BioNLI (Bastan et al., 2022), using three different prompts[6]

---

[5] Prompt is provided in App. A.1.6

[6] Prompts are shown in App. A.1.3, App. A.1.4, App. A.1.5.

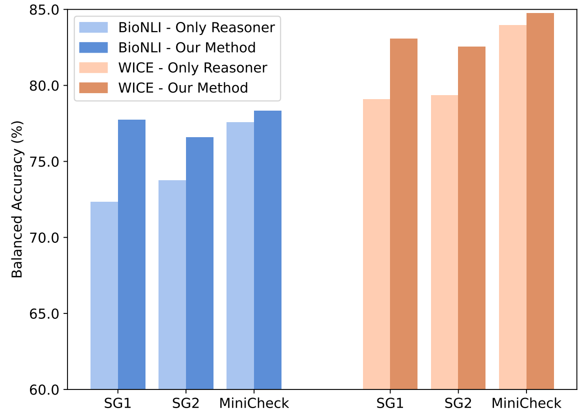


Figure 2: Balanced accuracy on the BioNLI (blue bar) and WiCE (orange bar) benchmarks under three prompt variants. Full results are reported in appendix (Fig. 4).

and three different models (`o4-mini`, *Qwen/QwQ-32B*, `Qwen/QwQ-32B`). Due to the high API cost of `o4-mini` and the large number of samples in the BioNLI dataset (Table 8), we used only the `Qwen` models[7] for evaluating the BioNLI full dataset.

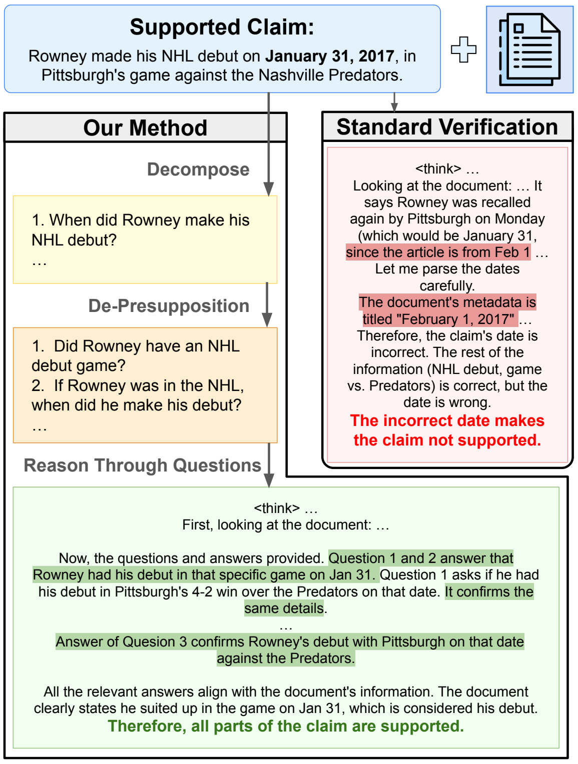
The results presented in Fig. 2 offer several key insights: (1) Our method outperforms the baseline across all datasets, prompts, and models, highlighting its effectiveness, and (2) It mitigates the performance degradation caused by prompt variance, demonstrating the robustness of the method.

3.4 Error Analysis

In Fig. 3 we present an excerpted annotated example from WiCE. On the left we show our approach, and on the right we show an LLM reasoner without our approach. Note that our structured reasoning can mitigate overthinking. In the red box, `Qwen3-32B` over-analyzed the evidence, placing unnecessary emphasis on the publication date (Feb 1) instead of the key event date (Jan 31), leading to an incorrect conclusion. In contrast, our method first decomposes the claim into high-level questions and then further into subatomic questions through de-presupposition, filtering out irrelevant information. We show additional full examples in the App. A.5.

4 Related Works

Many works have explored claim verification (Kamoi et al., 2023; Tang et al., 2024; Zha et al., 2023; Min et al., 2023; Song et al., 2024; Wang et al., 2024), but the decomposition of claims into questions is a relatively newer field. Chen et al. (2022) employed a trained question decomposer model to break down claims into multiple questions. In contrast, our method focuses on zero-shot question decomposition using the inherent knowledge embedded in LLMs, allowing it to be applied across different domains of claim verification, such as scientific, real-life, and political claims.



Rowney made his NHL debut on **January 31, 2017**, in Pittsburgh's game against the Nashville Predators.

1. When did Rowney make his NHL debut?
…

1. Did Rowney have an NHL debut game?
2. If Rowney was in the NHL, when did he make his debut?
…

<think> …
First, looking at the document: …

Now, the questions and answers provided. Question 1 and 2 answer that Rowney had his debut in that specific game on Jan 31. Question 1 asks if he had his debut in Pittsburgh's 4-2 win over the Predators on that date. It confirms the same details.
…
Answer of Quesion 3 confirms Rowney's debut with Pittsburgh on that date against the Predators.

All the relevant answers align with the document's information. The document clearly states he suited up in the game on Jan 31, which is considered his debut.
Therefore, all parts of the claim are supported.

<think> …
Looking at the document: … It says Rowney was recalled again by Pittsburgh on Monday (which would be January 31, since the article is from Feb 1 … Let me parse the dates carefully.
The document's metadata is titled "February 1, 2017" …
Therefore, the claim's date is incorrect. The rest of the information (NHL debut, game vs. Predators) is correct, but the date is wrong.
The incorrect date makes the claim not supported.

Figure 3: Question-answer-based reasoning (left) vs. typical verification by a reasoner model (right) on a SUPPORTED claim from WiCE (Kamoi et al., 2023).

Hu et al. (2025) demonstrated that direct claim decomposition yields mixed results depending on the strength of the verifier. Our findings are similar, with performance varying based on the type of prompts used. However, our work focuses on reducing this issue to a greater extent. Similarly, Fleischer et al. (2025) employed a question-based decomposition to answer the main question. Xue et al. applied a similar question-answer graph with a voting mechanism to solve mathematical problems. However, claim verification requires more extensive reasoning because its questions are less deterministic—unlike mathematical questions, claims do not follow a logical rationale, and thus the decomposition requires more filtering and reasoning.

In the multimodal domain, Cho et al. (2023) introduced a method of decomposing text into a graph of questions to evaluate text-to-image mod-

---

[7] Due to space, `o4-mini` results on the BioNLI-300 dataset are provided in Table 4 in the appendix.

els. Similarly, Jiang et al. (2024) decomposed both questions and images to answer the main question.

Recently, Lyu et al. (2025) has proposed a pipeline through question-answering to detect if a text has presupposition or not. Similarly, Kim et al. (2021) has shown that using presupposition-based decomposition improve the retrieval and hence improve the end-QA performance.

In contrast, our method focuses on a less explored aspect of decomposition, presupposition, and its impact on prompt variance and overthinking in reasoning models using dynamic question generation.

5 Conclusion

We introduce a novel question decomposition approach for generating presupposition-free, atomic questions that systematically interrogate each part of a claim. This study is inspired by how humans think by decomposing and verifying each part of a complex claim. Using our approach, we have shown that we can mitigate the prompt sensitivity by constraining it to thinking structurally than free form. We hope this line of research encourages further exploration into decomposition-based reasoning as a foundation for building more trustworthy and transparent claim verification systems.

Limitations

While the current LLMs are very effective as LLM-as-a-judge, there is always an inherent nature of randomness which can affect the question coverage metric. Also, we have not tested different other prompting methods due to the size of the dataset, computation constraints and high API cost. Furthermore, while we have shown empirically that our method improves the end-performance on claim verification task, we have not manually verified the outputs (if they are presupposition free) due to the size of the dataset.

Acknowledgments

We thank the reviewers for their detailed comments and suggestions. Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by DARPA for the SciFy program under agreement number HR00112520301. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA or the U.S. Government.

References

Mohaddeseh Bastan, Mihai Surdeanu, and Niranjan Balasubramanian. 2022. BioNLI: Generating a Biomedical NLI Dataset Using Lexico-semantic Constraints for Adversarial Examples. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5093–5104, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2023. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. *arXiv preprint arXiv:2310.18235*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2501.12948 [cs].

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. *arXiv preprint*. ArXiv:2012.00614 [cs].

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783.

Daniel Fleischer, Moshe Berchansky, Gad Markovits, and Moshe Wasserblat. 2025. SQuARE: Sequential Question Answering Reasoning Engine for Enhanced Chain-of-Thought in Large Language Models. *arXiv preprint*. ArXiv:2502.09390 [cs].

Qisheng Hu, Quanyu Long, and Wenya Wang. 2025. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6313–6336, Albuquerque, New Mexico. Association for Computational Linguistics.

Xinyi Jiang, Guoming Wang, Junhao Guo, Juncheng Li, Wenqiao Zhang, Rongxing Lu, and Siliang Tang. 2024. DIEM: Decomposition-Integration Enhancing Multimodal Insights. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27294–27303, Seattle, WA, USA. IEEE.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-World Entailment for Claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Najoung Kim, Ellie Pavlick, Burcu Karagol-Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Xiucheng Lyu, Runcong Zhao, Jiazheng Li, Bin Liang, Min Yang, Lin Gui, and Ruifeng Xu. 2025. Unsupervised fact-checking via recursively verifying presuppositions. *IEEE Transactions on Audio, Speech and Language Processing*, 33:2189–2199.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv preprint*. ArXiv:2305.14251 [cs].

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting. *arXiv preprint*. ArXiv:2310.11324 [cs].

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. *arXiv preprint*. ArXiv:2404.10774 [cs].

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint*. ArXiv:2307.09288 [cs].

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi Georgiev, Jiahui Geng, and Preslav Nakov. 2024. OpenFactCheck: Building, Benchmarking Customized Fact-Checking Systems and Evaluating the Factuality of Claims and LLMs. *arXiv preprint*. ArXiv:2405.05583 [cs].

Shangzi Xue, Zhenya Huang, Jiayu Liu, Binbin Jin, Xin Li, and Qi Liu. Decompose, Analyze and Rethink: Solving Intricate Problems with Human-like Reasoning Cycle.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating Factual Consistency with A Unified Alignment Function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976, Miami, Florida, USA. Association for Computational Linguistics.

A Appendix

A.1 Prompts

A.1.1 Question Decomposition

You are given a claim, your task is to decompose it into multiple independent and individual questions. DON'T generate any other text than the questions. You are given some examples below and the input claim at the end.

Claim: Other title changes included Lord Steven Regal and The Nasty Boys winning the World Television Championship and the World Tag Team Championship respectively.
Questions:
- Did Lord Steven Regal win the World Television Championship?
- Did The Nasty Boys win the World Tag Team Championship?

Claim: The parkway was opened in 2001 after just under a year of construction and almost two decades of community requests.
Questions:
- When was the parkway opened?
- How long was the construction period for the parkway?
- How many years of community requests preceded the opening of the parkway?

Claim: Touring began in Europe in April–June with guitarist Paul Gilbert as the opening act, followed by Australia and New Zealand in July, Mexico and South America in late July–August, and concluding in North America in October–November.
Questions:
- When did touring begin in Europe?
- Who was the opening act during the touring in Europe?
- Which months covered the Australia tour?
- Which months covered the New Zealand tour?
- Which months covered the Mexico tour?
- Which months covered the South America tour?
- Which months covered the North America tour?
- Where did the touring conclude?

Claim: In March 2018, the company partnered With Amazon Web Services (AWS) to offer Al-enabled conversational solutions to customers in India.
Questions:
- When did the company partner with AWS?
- What was the purpose of the partnership?

Claim: The most significant of these is in Germany, which now has a Yazidi community of more than 200,000 living primarily in Hannover, Bielefeld, Celle, Bremen, Bad Oeynhausen, Pforzheim and Oldenburg.
Questions:
- Which country hosts the largest Yazidi community?
- How large is the Yazidi community in Germany?
- In which cities are the Yazidi community in Germany primarily located?

Claim: A previous six-time winner of the Nations' Cup, Sebastian Vettel became Champion of Champions for the first time, defeating Tom Kristensen, who made the final for the fourth time, 2–0.
Questions:
- How many times had Sebastian Vettel won the Nations' Cup before?
- What title did Sebastian Vettel achieve for the first time?
- Whom did Sebastian Vettel defeat in the final?
- How many finals had Tom Kristensen reached?
- What was the final score between Sebastian Vettel and Tom Kristensen?

Claim: {{claim}}
Questions:

A.1.2 Question De-Presupposition

You are given a question that may contain presuppositions — assumptions that are implied but not necessarily true. Your task it to rewrite the question into one or more simpler de-contextualized questions that do not contain these presuppositions. DO NOT generate anything else other than the questions. You are also given some examples below and the input question at the end.

Question:
Which Bollywood movie has won the Oscar in 1928?

Rewritten questions:
- Was there an Oscar in 1928?
- If there was an Oscar in 1928, has any Bollywood movie won that?
- If any Bollywood movie won the Oscar in 1928, which one?

Question:
Which english movie was directed by Christopher Nolan?

Rewritten questions:
- Is Christopher Nolan a director?
- Has Christopher Nolan directed any english movie?
- If Christopher Nolan has directed any english movie, which one?

Question:
{{question}}

Rewritten questions:

A.1.3 Reasoner: SG1

You are an AI model tasked with verifying claims using zero-shot learning. Your job is to analyze a given claim along with provided evidence (i.e. corpus articles) and decide whether the available evidence would likely support or not support the claim. You are also given some questions that can help you analyze the claim and evidence.

Claim to evaluate:
<claim>
{{CLAIM}}
</claim>

Additional evidence provided:
<corpus_text>
{{EVIDENCE}}
</corpus_text>

Questions to consider:
<questions>
{{QUESTIONS}}
</questions>

Guidelines:
1. Evaluate the claim only based on the evidence provided.

2. It’s possible that you are given multiple evidence articles. It is also possible that some of the evidence articles are not relevant to the claim. Use your best judgement to determine which evidence to use and which to ignore.
3. Consider answering the questions one by one, before making a final decision.
4. If relevant information is not present in the evidence, then it is possible that the claim is not supported by the evidence. Use your best judgement and previous knowledge to make a decision.

After your analysis, output exactly one JSON object with exactly two keys: "reasoning" and "decision". The value associated with "decision" must be exactly one word – either "SUPPORTED" or "NOT_SUPPORTED" (uppercase, with no additional text). Do not add any markdown formatting, code fences, or additional text. " "The output must start with an opening curly brace {{ and end with a closing curly brace }}.

Example output format:
{{"reasoning": "Your brief explanation here (one or two sentences).", "decision": "SUPPORTED or NOT_SUPPORTED"}}

Now, please evaluate the above claim.

A.1.4 Reasoner: SG2

You are an AI model tasked with verifying claims using zero-shot learning. Your job is to analyze a given claim along with provided evidence and decide whether the available evidence would likely support or not support the claim. You are also given some questions that can help you analyze the claim and evidence.

### Instructions:
1. Evaluate the claim only based on the evidence provided.
2. Consider answering the questions one by one, before making a final decision.
3. It is possible that some of the questions are not relevant to the claim. Use your best judgement to determine which questions to answer and which to ignore.
4. Finally, analyze the claim, questions and evidence together and determine the label that best describes the relationship between the claim and the evidence.
5. The meaning of different labels:
- SUPPORTED: The claim is supported by the evidence.
- NOT_SUPPORTED: The claim is not supported by the evidence.

### Output Format:
After your analysis, output exactly one JSON object with exactly two keys: "reasoning" and "decision". The value associated with "decision" must be exactly one word – either "SUPPORTED" or "NOT_SUPPORTED" (uppercase, with no additional text). Do not add any markdown formatting, code fences, or additional text. " "The output must start with an opening curly brace {{ and end with a closing curly brace }}.

Example output format:
{{"reasoning": "Your brief explanation here (one or two sentences).", "decision": "SUPPORTED or NOT_SUPPORTED"}}

### Input:
Evidence:
{{EVIDENCE}}

Claim to evaluate:
{{CLAIM}}

Questions to consider:
{{QUESTIONS}}

### Output:

A.1.5 Reasoner: MiniCheck

Determine whether the provided claim is supported by the corresponding document. You are also given some decomposed questions derived from the claim. Reason through the questions to support your judgment. Support in this context implies that all information presented in the claim is substantiated by the document. If not, it should be considered not supported. Its possible that some of the questions are not relevant to the claim. Use your best judgement to determine which questions to consider and which to ignore. Fall back to the provided document when you are not sure about the question.

Document:
{{EVIDENCE}}

Claim:
{{CLAIM}}

Questions to consider:
{{QUESTIONS}}

Please assess the claim’s support with the document by responding with either "SUPPORTED" or "NOT_SUPPORTED". Do not generate anything else other than the answer.

Answer:

A.1.6 Question Coverage

To find out the coverage of the questions generated in the WiCE dataset, we have used the following prompt. The prompt has access to one subclaim and multiple questions and asks to find out if the subclaim is implicitly or explicitly covered by the questions. In the initial experiments, we have found out that given the evidence during the evaluation performs better due to the nominal references present in the subclaim or questions.

Given a claim, evidence, and a list of questions, analyze whether the questions collectively are sufficient to verify or refute the entire claim.

### Instructions
- We are looking for coverage of the claim not completeness of the questions. So, if some questions are not relevant to the claim, that’s fine. But if the relevant questions do not cover the whole claim, then the coverage is not good.
- The question does not need to ask the specific claim explicitly. If answering the question would verify the claim, then it covers the claim.

- It is possible that multiple questions together cover the claim. It is not necessary that the claim is covered by a single question.
- If a question and claim refer to similar, but non-identical concepts, use the provided evidence to determine whether the question and claim are referring to the same concept or not. For example, the claim may refer to "the machine learning technique," while the question may ask about "the supervised learning technique." Because the questions were generated based on the provided evidence, consider this evidence when determining your final answer.
- Begin by providing 1-2 sentences explaining your reasoning for the coverage of the claim.
- Afterward, output yes if the questions cover the claim completely, or no if they do not.
- Structure your final response into two sections:

  - EXPLANATION: (your reasoning in 1-2 sentences)

  - ANSWER: (Yes if the questions cover the claim completely, or No if they do not)

### Evidence
{{evidence}}

### Claim
{{claim}}

### Questions
{{questions}}

A.2 Experimental Setup

We have used `Qwen/QwQ-32B` and `o4-mini` as the reasoner models. For the QwQ model, we have used VLLM (Kwon et al., 2023) for inference (one H100) and the OpenAI API.Both models were run with the default temperature settings, while for `o4-mini`, we utilized the "high" reasoning setup. We performed three runs to report the mean performance with standard deviation for the BioNLI-300 and WiCE datasets. We conducted two runs for the BioNLI full dataset and `o4-mini` results due to high computation time and API costs.

A.3 Effect of Prompt Variance in Claim Verification

Sclar et al. (2024) found that even large models are susceptible to prompt variance, to confirm that on our task we have used 3 different prompts.

MiniCheck Prompt: We have adopted the Tang et al. (2024)'s verification prompt on this version. The exact prompt is reported on App. A.1.5.

Structured Guidance: To confirm the prompt variance, we have used 2 different structured guidance prompts with one minimal change – reorganization of the sections. Both prompts are reported on App. A.1.3 and App. A.1.4, both of those prompts were written by the authors and refined using the BioNLI validation set. We call them "SG1" and "SG2" respectively.

Prompt	BAcc	Supported	Refuted
BioNLI-300			
SG2	70.83 (-3.50)	87.33	54.33
SG1	68.33 (-6.00)	86.67	50.00
MiniCheck	74.33	74.00	74.67
WiCE			
SG2	79.35 (-4.60)	91.89	66.80
SG1	79.08 (-4.87)	94.59	63.56
MiniCheck	83.95	82.88	85.02

Table 3: Comparison of different prompts – SG1 - App. A.1.3, SG2 - App. A.1.4, MiniCheck - App. A.1.5. `o4-mini` with high-reasoning setup was used as reasoner (Average score is reported over two runs)

The results for different prompts are shown on the Table 3. The results show that the claim verification task, like many other tasks (Zhuo et al., 2024), is susceptible to the prompt (3-6%). The difference between the SG1 and SG2 is more shocking, as those two prompts are basically the same, with some minor reorganization of sections. Our method can effectively mitigate this prompt sensitivity of a reasoner model.

A.4 Additional Results

The full result of the experiment described in §3.3 is shown on the Table 4. In addition to the previous datasets, we have also provided results on the BioNLI-300 dataset to include the results for `o4-mini` as reasoner.

A.5 Qualitative Analysis

One of the success cases from the BioNLI dataset is shown in Table 6, while a failure case is presented in Table 7. The qualitative analysis reveals that, in some instances, the model overlooks portions of the decomposed questions (Table 7), leading to incorrect conclusions despite otherwise relevant evidence.

A.6 Dataset

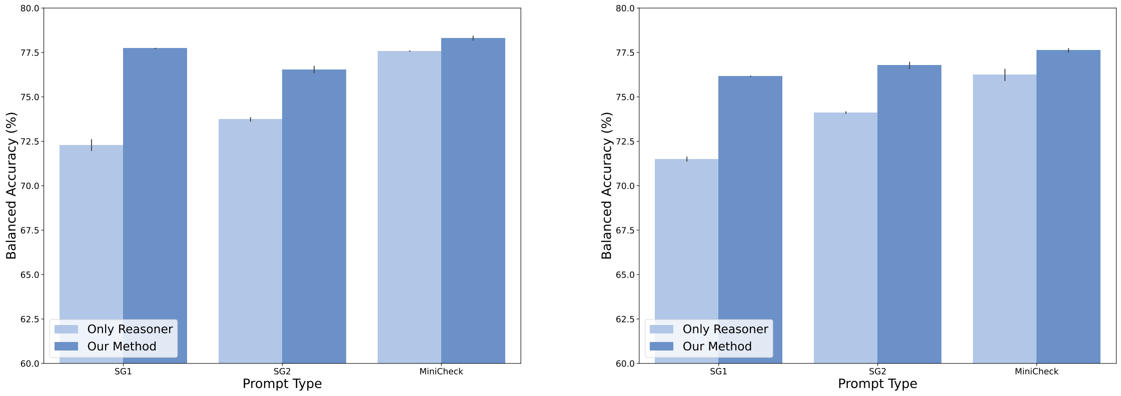
Full dataset statistics of WiCE, BioNLI and BioNLI-300 is provided on Table 8.

A.7 Results on Fever Dataset

We additionally report results on the FEVER test dataset (Thorne et al., 2018). Notably, FEVER was

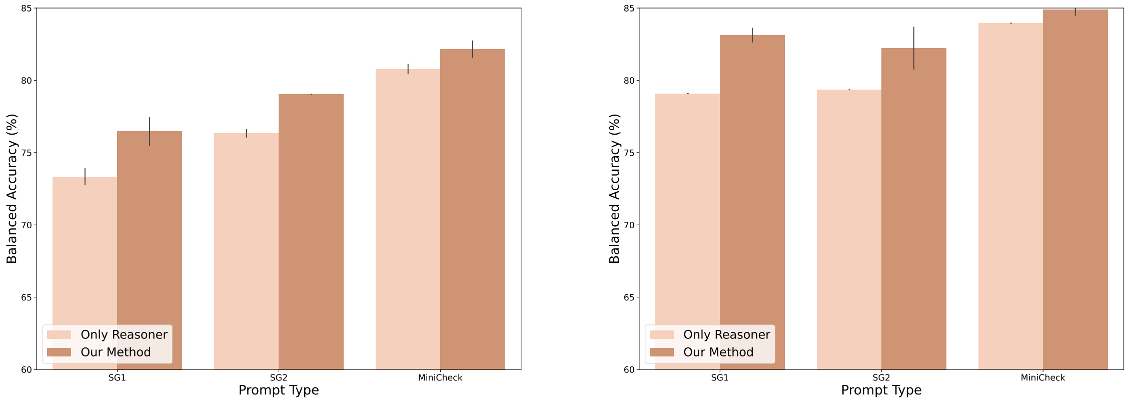
Prompt	Reasoner	Question Decomposer	De-Presupposition	BAcc	Supported	Refuted
BioNLI-FULL						
SG2	Qwen3-32B	Qwen3-32B	✓	76.79 ± 0.16	90.37 ± 0.72	63.21 ± 0.39
SG2	Qwen3-32B	Qwen3-32B	×	75.92 ± 0.16	89.83 ± 0.09	62.00 ± 0.42
SG2	Qwen3-32B	-	-	74.10 ± 0.04	89.65 ± 0.27	58.54 ± 0.19
SG2	QwQ-32B	QwQ-32B	✓	76.57 ± 0.19	90.10 ± 0.27	63.04 ± 0.11
SG2	QwQ-32B	QwQ-32B	×	76.57 ± 0.08	89.92 ± 0.09	63.23 ± 0.08
SG2	QwQ-32B	-	-	73.74 ± 0.10	91.49 ± 0.14	55.99 ± 0.06
SG1	Qwen3-32B	Qwen3-32B	✓	76.16 ± 0.01	87.49 ± 0.18	64.84 ± 0.15
SG1	Qwen3-32B	Qwen3-32B	×	75.38 ± 0.31	87.76 ± 0.72	63.00 ± 0.10
SG1	Qwen3-32B	-	-	71.49 ± 0.11	89.87 ± 0.41	53.10 ± 0.18
SG1	QwQ-32B	QwQ-32B	✓	77.73 ± 0.06	87.04 ± 0.00	68.41 ± 0.11
SG1	QwQ-32B	QwQ-32B	×	76.72 ± 0.14	87.98 ± 0.05	65.46 ± 0.32
SG1	QwQ-32B	-	-	72.34 ± 0.26	93.92 ± 0.32	50.76 ± 0.20
MiniCheck	Qwen3-32B	Qwen3-32B	✓	77.60 ± 0.10	81.14 ± 0.14	74.05 ± 0.33
MiniCheck	Qwen3-32B	Qwen3-32B	×	76.34 ± 0.32	80.65 ± 0.54	72.03 ± 0.10
MiniCheck	Qwen3-32B	-	-	76.29 ± 0.28	80.11 ± 0.09	72.46 ± 0.48
MiniCheck	QwQ-32B	QwQ-32B	✓	78.32 ± 0.15	86.45 ± 0.14	70.18 ± 0.16
MiniCheck	QwQ-32B	QwQ-32B	×	78.04 ± 0.14	86.54 ± 0.32	69.54 ± 0.03
MiniCheck	QwQ-32B	-	-	77.58 ± 0.14	84.92 ± 0.23	70.23 ± 0.06
BioNLI-300						
SG2	o4-mini	o4-mini	✓	74.83 ± 0.50	85.00 ± 1.00	64.67 ± 0.00
SG2	o4-mini	o4-mini	×	71.33 ± 0.33	85.67 ± 0.33	57.00 ± 1.00
SG2	o4-mini	-	-	70.83 ± 0.17	87.33 ± 0.00	54.33 ± 0.33
SG2	QwQ-32B	QwQ-32B	✓	73.44 ± 1.40	85.56 ± 0.83	61.33 ± 1.96
SG2	QwQ-32B	QwQ-32B	×	72.56 ± 1.50	86.44 ± 1.13	58.67 ± 1.96
SG2	QwQ-32B	-	-	69.11 ± 0.42	88.00 ± 0.54	50.22 ± 0.31
SG1	o4-mini	o4-mini	✓	73.00 ± 0.33	82.33 ± 1.00	63.67 ± 0.33
SG1	o4-mini	o4-mini	×	71.67 ± 0.00	83.33 ± 0.67	60.00 ± 0.67
SG1	o4-mini	-	-	68.33 ± 0.00	86.67 ± 0.00	50.00 ± 0.00
SG1	QwQ-32B	QwQ-32B	✓	75.00 ± 2.18	86.00 ± 0.54	64.00 ± 3.81
SG1	QwQ-32B	QwQ-32B	×	74.11 ± 1.29	86.44 ± 0.83	61.78 ± 1.75
SG1	QwQ-32B	-	-	68.44 ± 0.68	92.44 ± 0.83	44.44 ± 0.63
MiniCheck	o4-mini	o4-mini	✓	74.67 ± 0.33	79.00 ± 1.00	70.33 ± 0.33
MiniCheck	o4-mini	o4-mini	×	71.83 ± 0.17	79.00 ± 0.33	64.67 ± 0.00
MiniCheck	o4-mini	-	-	74.33 ± 0.00	74.00 ± 0.00	74.67 ± 0.00
MiniCheck	QwQ-32B	QwQ-32B	✓	75.11 ± 1.10	82.44 ± 0.63	67.78 ± 1.57
MiniCheck	QwQ-32B	QwQ-32B	×	74.33 ± 1.25	83.33 ± 1.63	65.33 ± 0.94
MiniCheck	QwQ-32B	-	-	73.33 ± 0.98	82.67 ± 1.09	64.00 ± 1.44
WiCE						
SG2	o4-mini	o4-mini	✓	82.54 ± 1.46	90.99 ± 0.90	74.09 ± 2.02
SG2	o4-mini	o4-mini	×	83.32 ± 0.17	92.34 ± 1.35	74.29 ± 1.01
SG2	o4-mini	-	-	79.35 ± 0.00	91.89 ± 0.00	66.80 ± 0.00
SG2	QwQ-32B	QwQ-32B	✓	79.03 ± 0.00	85.59 ± 0.00	72.47 ± 0.00
SG2	QwQ-32B	QwQ-32B	×	79.32 ± 0.49	84.68 ± 0.74	73.95 ± 1.38
SG2	QwQ-32B	-	-	76.36 ± 0.27	86.19 ± 0.42	66.53 ± 0.19
SG1	o4-mini	o4-mini	✓	83.07 ± 0.53	91.44 ± 0.45	74.70 ± 0.61
SG1	o4-mini	o4-mini	×	83.52 ± 0.33	92.34 ± 0.45	74.70 ± 0.20
SG1	o4-mini	-	-	79.08 ± 0.00	94.59 ± 0.00	63.56 ± 0.00
SG1	QwQ-32B	QwQ-32B	✓	76.42 ± 0.92	83.48 ± 1.12	69.37 ± 1.99
SG1	QwQ-32B	QwQ-32B	×	78.23 ± 0.64	85.89 ± 0.42	70.58 ± 1.66
SG1	QwQ-32B	-	-	73.41 ± 0.56	91.89 ± 0.74	54.93 ± 0.69
MiniCheck	o4-mini	o4-mini	✓	84.74 ± 0.40	86.49 ± 0.00	83.00 ± 0.81
MiniCheck	o4-mini	o4-mini	×	84.09 ± 0.86	85.59 ± 0.90	82.59 ± 0.81
MiniCheck	o4-mini	-	-	83.95 ± 0.00	82.88 ± 0.00	85.02 ± 0.00
MiniCheck	QwQ-32B	QwQ-32B	✓	82.25 ± 0.62	82.58 ± 1.12	81.92 ± 0.83
MiniCheck	QwQ-32B	QwQ-32B	×	81.72 ± 0.50	82.88 ± 1.27	80.57 ± 0.33
MiniCheck	QwQ-32B	-	-	80.70 ± 0.30	82.58 ± 0.42	78.81 ± 0.50

Table 4: Detailed Results on the WiCE, BioNLI Full and BioNLI-300 dataset.



(a) BioNLI dataset with `Qwen/QwQ-32B` Reasoner and Question-decomposer

(b) BioNLI dataset with `Qwen3-32B` Reasoner and Question-decomposer



(c) WiCE dataset with `Qwen/QwQ-32B` Reasoner and Question-decomposer

(d) WiCE dataset with `o4-mini` Reasoner and Question-decomposer

Figure 4: Balanced accuracy on the BioNLI (a, b) and WiCE (c, d) benchmarks under three prompt variants. Across each dataset, prompt variant, and backbone LLM, our method yields consistent and significant gains over the single-reasoner baseline.

Prompt	De-Presupposition	Answer Module	BAcc
MiniCheck	×	✓	71.33 ± 0.00
MiniCheck	×	×	71.83 ± 0.17
MiniCheck	✓	✓	73.50 ± 0.17
MiniCheck	✓	×	74.67 ± 0.33
SG1	×	✓	69.83 ± 0.83
SG1	×	×	71.67 ± 0.00
SG1	✓	✓	73.33 ± 0.67
SG1	✓	×	73.00 ± 0.33
SG2	×	✓	69.67 ± 0.67
SG2	×	×	71.33 ± 0.33
SG2	✓	✓	71.67 ± 1.00
SG2	✓	×	74.83 ± 0.50

Table 5: Results (with standard deviation) of using explicit answer module with our method.

originally developed as a retrieval and coarse verification benchmark rather than a fine-grained verification task. For example, one of the test claims– “Hearts is a musical composition by Minogue”– lacks the complexity necessary for decomposition-based reasoning. Furthermore, as shown in Table 8, the average claim length is only 8 words, indicating that the claims are already highly atomic and offer limited opportunity for further decomposition.

As shown in Table 9, our decomposition and de-presupposition pipeline does not yield significant improvements on this dataset, highlighting the importance of task complexity needed in benefiting from structured reasoning approaches. Unlike FEVER, most claims in real-world and scientific domains are highly complex, often requiring multi-hop reasoning. This underscores the necessity of decomposition-based approaches for effective claim verification in such settings.

Label	Content
Claim	We conclude that the reduction in furosemide stimulated renin release by indomethacin is due to renal cyclo-oxygenase inhibition which is reflected in decreased excretion rates of hydrolysis products of renal eicosanoids.
Decomposed Questions	1: What is the cause of the reduction in furosemide-stimulated renin release? 2: What is the effect of indomethacin on furosemide-stimulated renin release? 3: What is reflected in decreased excretion rates of hydrolysis products of renal eicosanoids?
Presupposition free Questions	1: If the reduction in furosemide-stimulated renin release has a cause, what is the cause? 2: What is the effect of indomethacin on furosemide-stimulated renin release? 3: If so, has anything been reflected in those decreased excretion rates? 4: What is reflected in decreased excretion rates of hydrolysis products of renal eicosanoids? 5: If indomethacin has an effect on furosemide-stimulated renin release, what is the effect? 6: Does furosemide stimulate renin release? 7: If there is a reduction in furosemide-stimulated renin release, does it have a cause? 8: If yes, what is reflected in them? 9: What is the cause of the reduction in furosemide-stimulated renin release? 10: If furosemide stimulates renin release, does indomethacin have an effect on that release? 11: Do hydrolysis products of renal eicosanoids have excretion rates that can decrease? 12: Is there a reduction in furosemide-stimulated renin release?
Reasoning	First, I need to check the document for evidence supporting each part of the claim. **Looking at the document:** The study involved giving indomethacin to normal volunteers and then measuring responses to furosemide. The key points from the document are: - Indomethacin reduced serum thromboxane B2 (TXB2), which is a platelet product, but that's not directly about the kidney. However, the document mentions that renal prostacyclin and thromboxane A2 synthesis are limited to the cortex. So maybe the eicosanoids in the kidney are affected too. - The study found that both basal and furosemide-stimulated plasma renin activity were reduced after indomethacin. Also, the excretion rates of 6-keto-PGF1 alpha (a prostacyclin metabolite) and TXB2 (though TXB2 here might be from the kidney, but earlier they said TXB2 is platelet-derived, so maybe the renal TXB2 is different? Wait, the document says "the transient increase in excretion rates of 6-keto-PGF1 alpha and TXB2" were also reduced. So maybe the TXB2 here refers to renal, not platelet? Hmm, that's a bit confusing because earlier they said serum TXB2 was from platelets. Maybe the excretion in urine includes renal sources? The document says "excretion rates" so that's probably urine. So the renal eicosanoids' hydrolysis products would be in urine. The claim states that the decrease in renin release is due to cyclo-oxygenase inhibition (which indomethacin does) and this is reflected in decreased excretion of hydrolysis products. The document does mention that after indomethacin, the excretion of 6-keto-PGF1 alpha and TXB2 (the hydrolysis products) decreased, along with the renin. Since prostacyclin and TXA2 are eicosanoids made via cyclo-oxygenase, their hydrolysis products being reduced would indicate that indomethacin (a COX inhibitor) is blocking their synthesis. Thus, the reduced renin release is linked to the inhibition of these eicosanoids, which are part of renal cyclo-oxygenase products. **Looking at the questions:** 1. The cause of the reduction in renin release would be indomethacin's effect via COX inhibition. The document says that after indomethacin, renin is reduced, so that's supported. 2. Indomethacin reduces furosemide-stimulated renin release (document says "reduced, as well as the transient increase..."), so yes. 3. The decreased excretion rates would reflect the inhibition of eicosanoid synthesis. The document mentions decreased excretion of 6-keto-PGF1 and TXB2, so that's the hydrolysis products. 4. The reflection is the renal cyclo-oxygenase inhibition. Since those products are from renal eicosanoids (as per the first sentence about their synthesis being limited to the cortex), their decreased excretion would reflect that inhibition. 5. The effect is reduction of renin release, as in question 2. 6. The document states that furosemide does stimulate renin, as they mention "furosemide stimulated plasma renin activity". 7. The reduction does have a cause (indomethacin's COX inhibition). 8. The decreased excretion rates reflect the inhibition of eicosanoid synthesis. 9. The cause is indomethacin's cyclo-oxygenase inhibition, which the document implies since indomethacin is a COX inhibitor and the eicosanoid products are reduced. 10. Yes, indomethacin reduces the renin release from furosemide. 11. Yes, the document says excretion rates of those products decreased. 12. Yes, the study found a reduction. **Putting it all together:** The claim's main points are that the renin reduction is due to COX inhibition (indomethacin) and that this is shown by decreased excretion of hydrolysis products. The document supports both the reduction in renin and the decrease in the excretion of those products (6-keto-PGF1 alpha and TXB2) after indomethacin. Since cyclo-oxygenase inhibition by indomethacin would block the synthesis of prostacyclin and thromboxane, leading to less of their hydrolysis products being excreted, the reasoning in the claim is supported by the document's findings. Therefore, the answer should be **SUPPORTED**.

Table 6: One of the examples from the BioNLI dataset where the reasoner correctly verified the claim as SUPPORTED.

Label	Content
Claim	We conclude that ACTH hypersecretion after CRH-mediated stimuli, including naloxone, is an inherent, but variable, feature of DM, caused by expression of the genetic mutation at the anterior pituitary.
Decomposed Questions	1: Where is the genetic mutation expressed that leads to ACTH hypersecretion in DM? 2: What is an example of a CRH-mediated stimulus mentioned in the claim? 3: What causes the ACTH hypersecretion in DM? 4: What type of stimuli trigger ACTH hypersecretion in DM? 5: What is an inherent but variable feature of DM?
Presupposition-free Questions	1: If DM exists, does it have any features? 2: If there are stimuli that trigger ACTH hypersecretion in DM, what type are they? 3: If DM experiences ACTH hypersecretion, are there any stimuli that trigger it? 4: If ACTH hypersecretion occurs in DM, what causes it? 5: What is an example of a CRH-mediated stimulus mentioned in the claim? 6: If DM has features, does it have any inherent features? 7: What type of stimuli trigger ACTH hypersecretion in DM? 8: If DM exists, does ACTH hypersecretion occur in it? 9: If DM has an inherent but variable feature, which one is it? 10: If DM has inherent features, are any of them variable? 11: What causes the ACTH hypersecretion in DM? 12: Does DM experience ACTH hypersecretion? 13: If such a mutation exists, is it expressed in a specific location? 14: Does DM exist? 15: Where is the genetic mutation expressed that leads to ACTH hypersecretion in DM? 16: If the claim mentions any CRH-mediated stimuli, what is an example? 17: Does a genetic mutation lead to ACTH hypersecretion in DM? 18: If it is expressed in a specific location, where is that? 19: What is an inherent but variable feature of DM? 20: Does the claim mention any CRH-mediated stimuli?
Reasoning	First, **I'll look at the document.** The study shows that in DM patients, their basal ACTH levels are twice as high as controls, and their ACTH responses to naloxone (a CRH stimulator) are four times higher. The cortisol responses were similar, but ACTH was definitely higher. They also mention that 19 out of 36 patients were hyperresponders, and some had reproducible responses. **Now, looking at the questions:** 1. Does DM have features? The document does talk about ACTH hypersecretion as a feature, so yes. 2. The stimuli are CRH-mediated, like naloxone. The document mentions naloxone as a CRH stimulator. 3. Yes, CRH-mediated stimuli trigger ACTH hypersecretion in DM. 4. The cause according to the claim is the genetic mutation in the anterior pituitary. The document says they "previously showed" CRH-mediated stimuli cause this, but does it mention the genetic mutation's role? The document here doesn't explicitly state the genetic mutation's location. Wait, the claim says it's caused by the mutation's expression in the anterior pituitary. The document's current evidence is about the response to naloxone and ACTH levels but doesn't mention the genetic mutation's expression site. Hmm, that's a problem. The study might be part of their previous work, but the current document doesn't specify the cause as the genetic mutation in the anterior pituitary. Questions 5-12 are missing Looking at the questions 13,14,15,17,18: The claim says the cause is the genetic mutation expressed in the anterior pituitary. The document here doesn't mention the location of the mutation's expression. The study talks about the effect (ACTH levels) but not the exact mechanism's location. So the part about the anterior pituitary might not be supported by the provided document. Questions 19,20 are missing The claim also says it's an inherent feature. The document does say that DM patients have higher ACTH responses consistently, even in some repeated tests, so that's inherent. The variability is mentioned with hyperresponders (19 out of 36), so variable. But the key issue is the cause being the genetic mutation in the anterior pituitary. The document's current text doesn't state that. The first sentence says they "previously showed" CRH-mediated stimuli cause ACTH hypersecretion. Maybe the prior work mentioned the genetic cause, but in the provided document, the cause isn't specified beyond the stimuli leading to the response. The conclusion in the claim adds the genetic mutation's role in the anterior pituitary, which isn't present here. Therefore, even though most parts of the claim are supported (CRH-mediated stimuli like naloxone cause ACTH hypersecretion, inherent and variable), the specific part about the genetic mutation's expression in the anterior pituitary isn't addressed in the provided document. So the entire claim **isn't fully supported** because that part isn't there.

Table 7: One of the examples from the BioNLI dataset where the reasoner incorrectly verified the claim as NOT-SUPPORTED though the claim is originally SUPPORTED.

Dataset	Samples	Supported	Refuted	Average # of Words	
				Claim	Evidence
WiCE (Kwon et al., 2023)	358	247	111	24	1316
BioNLI (Bastan et al., 2022)	5073	3962	1111	34	187
BioNLI-300 (sampled)	300	150	150	35	185
FEVER (Diggelmann et al., 2021)	6605	3305	3300	8	305

Table 8: Statistics of different dataset used in the study. Following previous papers (Tang et al., 2024; Kamoi et al., 2023), we have converted the WiCE dataset to binary by assuming partially supported as refuted.

Prompt	Reasoner	Question Decomposer	De-Presupposition	BAcc	Supported	Refuted
SG2	Qwen3-32B	Qwen3-32B	✓	95.27 ± 0.07	93.90 ± 0.23	96.64 ± 0.09
SG2	Qwen3-32B	Qwen3-32B	×	95.49 ± 0.11	94.55 ± 0.18	96.42 ± 0.03
SG2	Qwen3-32B	-	×	95.47 ± 0.04	94.49 ± 0.00	96.44 ± 0.08
SG2	QwQ-32B	QwQ-32B	✓	95.44 ± 0.17	94.25 ± 0.24	96.62 ± 0.11
SG2	QwQ-32B	QwQ-32B	×	95.42 ± 0.07	94.24 ± 0.05	96.61 ± 0.09
SG2	QwQ-32B	-	×	95.56 ± 0.02	94.70 ± 0.09	96.42 ± 0.06
SG1	Qwen3-32B	Qwen3-32B	✓	95.18 ± 0.17	93.92 ± 0.18	96.44 ± 0.17
SG1	Qwen3-32B	Qwen3-32B	×	95.28 ± 0.03	94.22 ± 0.06	96.33 ± 0.00
SG1	Qwen3-32B	-	×	95.62 ± 0.05	95.08 ± 0.08	96.17 ± 0.02
SG1	QwQ-32B	QwQ-32B	✓	95.31 ± 0.08	94.30 ± 0.08	96.33 ± 0.09
SG1	QwQ-32B	QwQ-32B	×	95.49 ± 0.06	94.58 ± 0.09	96.39 ± 0.03
SG1	QwQ-32B	-	×	95.64 ± 0.05	95.40 ± 0.00	95.88 ± 0.09
MiniCheck	Qwen3-32B	Qwen3-32B	✓	95.00 ± 0.11	93.22 ± 0.21	96.77 ± 0.02
MiniCheck	Qwen3-32B	Qwen3-32B	×	95.36 ± 0.08	94.21 ± 0.11	96.52 ± 0.06
MiniCheck	Qwen3-32B	-	×	95.41 ± 0.01	93.96 ± 0.11	96.85 ± 0.09
MiniCheck	QwQ-32B	QwQ-32B	✓	95.40 ± 0.02	94.13 ± 0.00	96.67 ± 0.03
MiniCheck	QwQ-32B	QwQ-32B	×	95.47 ± 0.05	94.40 ± 0.03	96.53 ± 0.08
MiniCheck	QwQ-32B	-	×	95.40 ± 0.03	94.16 ± 0.06	96.64 ± 0.00

Table 9: Results (with standard deviation) on the FEVER (Thorne et al., 2018) dataset. Due to the size of the dataset, we have reported scores over two runs.

Modeling Language Learning in Corrective Feedback Interactions

Juan L. Castro-Garcia
Computer Science and Engineering
Michigan State University
castrog4@msu.edu

Parisa Kordjamshidi
Computer Science and Engineering
Michigan State University
kordjams@msu.edu

Abstract

To study computational models for language acquisition, we propose an interactive computational framework that utilizes a miniature language acquisition dataset in a controlled environment. In this framework, a neural learner model interacts with a teacher model that provides corrective feedback. Within this framework, we investigate various corrective feedback strategies, specifically focusing on reformulations and their effect on the learner model during their interactions. We design experimental settings to evaluate the learner's production of syntactically and semantically correct linguistic utterances and perception of concepts and word-meaning associations. These results offer insights into the effectiveness of different feedback strategies in language acquisition using artificial neural networks. The outcome of this research is establishing a framework with a dataset for the systematic evaluation of various aspects of language acquisition in a controlled environment.

1 Introduction

Understanding how children form associations between linguistic words to some situational input or referent within an uncertain environment where multiple referents could be related to the same word is a topic that has been studied in language acquisition (Quine, 1960). Cross-situational learning is a powerful mechanism for learning co-occurrence statistics between words and referent objects across multiple exposures (Gleitman, 1990; Pinker, 2013). Studies of cross-situational learning in both adults (Yu and Smith, 2007; Smith et al., 2011) and children (Suanda et al., 2014; Smith and Yu, 2008) show how the association between words and meaning is learned at different stages of language development. However, some of these studies on cross-situational learning focus on the child's learning of these word-meaning associations without any form of feedback (Monaghan et al., 2021).

Feedback, in the form of social interactions, is shown to enhance children's language development (Kuhl et al., 2003; Sachs et al., 1981; Krashen et al., 1983). In language acquisition studies, most commonly in second language acquisition literature, an interaction is viewed as a negotiation for meaning where two agents "negotiate" or agree upon the meaning of some object during a conversation (Long, 1981; Clark, 1996). Corrective Feedback is one form of interaction where an adult, i.e. parent or teacher, analyzes the linguistic generation of a child and provides some form of response intended to adjust or update the child's linguistic knowledge. Although the impact of providing corrective feedback is a controversial topic, many studies supports its influence on language learning even in first language acquisition (Hiller, 2016; Chouinard and Clark, 2003; Schoneberger, 2010). In a social context, commonly within a classroom setting, several approaches for corrective feedback are utilized such as explicit correction, recast or reformulations, clarification request, metalinguistic feedback, elicitation, and repetition (Lyster and Ranta, 1997). For the scope of this paper, we will focus on reformulations as our computational approach to corrective feedback.

Cross-situational learning has been used to address multiple tasks like probabilistic word-meaning learning with symbolic situation representations (Fazly et al., 2010), word-meaning learning with embodied systems (Yu and Ballard, 2004), and word-meaning associations from visual perceptual representations as inputs (Juven and Hinaut, 2020). Also, several studies have explored learning settings that simulate interactions between teacher and learner conversations with k-Nearest Neighbor models for word learning (Belpaeme and Morse, 2012) and probabilistic models with corrections (Angluin and Becerra-Bonache, 2017). Other models have studied the acquisition of semantic knowledge through some form of feedback provided via a reward function in a reinforcement learning setting (Nikolaus and Fourtassi, 2021b,a). Other studies have explored the language acquisition process as a game where two agents observe referents in a scene and both attempt to name it (Steels, 1995).

Building neural computational models for Language Acquisition based has been practiced in the related literature (Portelance and Jasbi, 2023; Frank et al., 2019). However, one issue that these models present is the requirement of large amount of linguistic data, most times larger than what humans are exposed to through out their entire lifetime. Also, the architecture design of these models can have innate biases that makes relating their outcome analysis to human language acquisition theories challenging (Baroni, 2022). In order to

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 267–279
November 8-9, 2025 ©2025 Association for Computational Linguistics

establish a fair and accurate comparison between neural models and human learners, it is required to simulate learning scenarios where the quantity of data, input modalities, and data distributions resemble human-level abilities (Warstadt and Bowman, 2024).

In this paper, we use a learning scenario that explores neural computational models for language acquisition in a teacher-learner interactive cross-situational learning framework. This framework follows a similar structure as the one presented in Angluin and Becerra-Bonache (2017) which uses a Miniature Language Acquisition dataset where the environment contains logical representation of objects and the model learns to generate linguistic utterances that describe these objects. We extend their work by using recurrent neural networks as the learner rather than a probabilistic graphical model. We train small models from scratch, without any prior knowledge, that resembles early stages of concept learning and language acquisition.

Also, we examine various corrective feedback strategies and their impact on the learner model's learning trajectory. The learner is evaluated at a production level based on utterance semantics and its ability to generate all possible explanations, as well as a perception level, referring to the knowledge acquired about concepts and relationships. Figure 1 shows how interaction is established, where each interaction starts with a learner model generating syntactically and semantically appropriate utterances corresponding to the attributes and relationships within a given situation, represented formally in a formal predicate-argument form. The teacher then analyzes the utterance, compares the utterance's formal representation to the situation's formal representation, and provides another utterance with a similar formal representation to the situation.

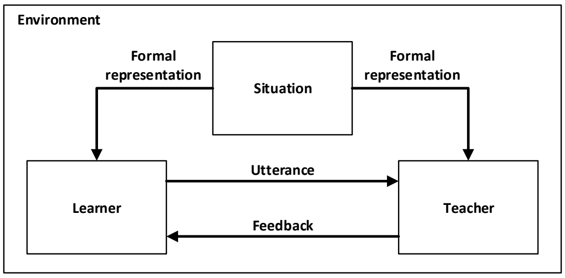


Figure 1: Teacher-Learner Interaction model. This framework has a teacher and learner model interact in an environment with shared situations. The learner model attempts to describe the situation, and then the teacher evaluates the description and provides feedback to the learner.

The main contributions of this work are as follows: 1) Extending Miniature Language Acquisition computational framework with neural learner models; 2) Providing evaluation datasets and metrics for utterance generation (production-level) and concept evaluation (perception-level); 3) Incorporating various corrective feedback strategies in the form of interactions with an oracle that evaluates the logical semantics.

2 Miniature Language Acquisition Dataset

Miniature Language Acquisition (MiLA) is a task that consists on the learning of a natural language from sentence-picture pairs, where each "picture" or scene is composed of geometric shapes with different properties (Feldman et al., 1990). Similar to the setup in (Angluin and Becerra-Bonache, 2017), we create a learning setting in which the learner receives a formal representation of the environment, instead of actual visual input, and generates natural language utterances that explain the environment. In our experiments, we only use English natural language utterances. The dataset includes pairs of formal representations of various situations with their respective set of valid linguistic utterances. There are 23,328 unique situations, where each situation refers to two objects with all three attributes and relates to 113,064 unique utterances. Every situation is paired with at most 40 utterances.

2.1 Situation Representation

We define a formal representation of objects, their properties, and their relations within the environment using a predicate-argument structure referred to as a situation s. The s represents the full formal representation of the situation but we also use a partial one referred to as m. Each predicate p can have one or two arguments representing objects, denoted by t. For example, $p(t_1)$ represents a single-argument predicate, and $p(t_1, t_2)$ represents a two-argument predicate.

Single-argument predicates describe the properties of an object including $\{shape, size, color\}$ whereas two-argument predicates describe the relationships between objects including $\{left, above\}$. Shape predicates include hexagon ($he1$), star ($st1$), triangle ($tr1$), square ($sq1$), circle ($ci1$), and ellipse ($el1$). Colors include red ($re1$), blue ($bl1$), yellow ($ye1$), orange ($or1$), green ($gr1$), and purple ($pu1$). Sizes include small ($sm1$), medium ($me1$), and big ($bi1$). Relations include left ($le2$) and above ($ab2$). Although the relation predicates are limited to two relation types, these are sufficiently expressive for two additional relations: right and below. For utterances that include right and below, their formal situation will use the predicates $le2$ and $ab2$ respectively while the order of their arguments reflects the actual relationship.

Each *full situation* in the dataset includes all the properties of two objects and their relation. A partial situation representation can include a subset of the predicates. However, a valid partial situation must have at least one shape predicate to be able to refer to at least one object. We refer to the object types, attributes, and their relations as concepts to be learned. We generate all possible formal situations based on the possible combinations of the concepts given a set of templates. An example of a formal situation is:

```
sm1(t1),bl1(t1),ci1(t1),
ab2(t1,t2),
bi1(t2),re1(t2),sq1(t2)
```

where the following natural language expression, “the small blue circle above the big red square” is a valid explanation for it. In the generation process of formal situation representation, we consistently use the order of object/relation/object where each object is described as follows: size/color/shape.

2.2 Linguistic Utterances

A linguistic utterance, denoted as u, is a sequence of words used to describe a formal representation of situations mentioned in Section 2.1. All linguistic utterances in this data follow the grammar shown in Figure 2 with a vocabulary size of over 20 words. This grammar generate utterances like: “the small blue circle”, “the star”, and “the small yellow square above the medium red hexagon”.

S → <Object> | <Object> <UpDown> <Object> | <Object> to the <LeftRight> of <Object>
Object → the <Size><Color><Shape>
Size → big | medium | small | ϵ
Color → red | blue | yellow | orange | green | purple | ϵ
Shape → circle | triangle | square | hexagon | star | ellipse
UpDown → above | below
LeftRight → right | left

Figure 2: Grammar used for utterance generation.

2.3 Situation-Utterances Alignment

To connect all valid utterances from the grammar shown in Section 2.2 to a formal situation mentioned in Section 2.1, we define a meaning transducer T which receives an utterance u as input and generates a formal representation of the utterance m. An example of this transducer T is shown in Figure 3. Table 1 shows examples of linguistic utterances and their formal representations. For example, the utterance “the small circle” is mapped to shape and size predicates with the following predicates “$sm1(t_1), ci1(t_1)$”, which is a valid formal representation generated by the transducer T. In addition of building the dataset for the experiments presented in this paper, the teacher model uses this transducer T twofold: (1) determine if the utterance follows the grammar and (2) generates the formal representation m from the learner utterance for feedback generation.

3 Language Acquisition Setup

The language acquisition framework presented here aims to evaluate the influence of corrective feedback in the development of a learner model’s capacity to describe various situations as well as its association of each word to a corresponding predicate. The learner receives formal situation representations which it uses to produce a valid utterance that describes the situation. The teacher model employs various feedback strategies to choose an utterance from a set of valid utterances in order to address any possible errors that said utterance might have. Figure 4 shows an interaction between the learner and the teacher.

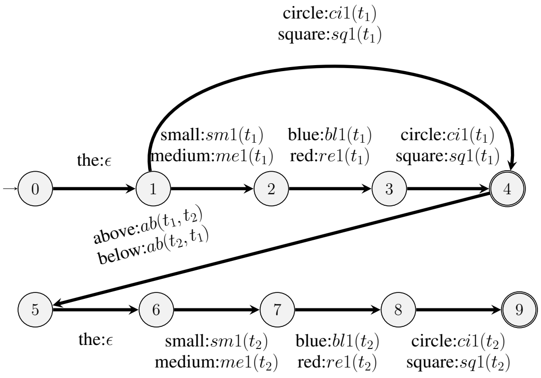


Figure 3: Meaning Transducer T. This is an example of the transducer which generates the formal representation of any utterance from the grammar.

3.1 Learner Model

The learner model is implemented as an Encoder-Decoder model with Gated Recurrent Units (GRU) (Cho et al., 2014; Sutskever et al., 2014) that receives a situation s which is a sequence of predicates as input and generates an utterance u which is a sequence of words describing the situation. We incorporate an attention module in the GRU architecture to improve learning the association between words and predicates (Bahdanau et al., 2016). The learner model uses cross-entropy (CE) loss between the learner utterance and the perceived utterances which are computed as follows:

$$l = -\sum_{t=1}^{T} \log p_w(y_t | u_{<t}, s) \qquad (1)$$

where $y = y_1, \cdots, y_t$ is the ground truth utterance (which is selected based on the feedback strategy), u is the learner-generated utterance, and p_w, is the probability for generating the utterance given the situation parameterized with w.

3.2 Teacher Model

The teacher in this experiment is not a neural model rather is a predefined evaluator composed of an Evaluation and Feedback Generation modules. The former acquires the logical semantics of an utterance using the transducer defined in Section 2.3 to determine the validity of the utterance. The latter uses the logical semantics of the utterance and the situation to select a

Linguistic representation (u)	Formal representation (m)
“the small circle”	$sm1(t_1), ci1(t_1)$
“the small blue circle to the left of the big red square”	$sm1(t_1), bl1(t_1), ci1(t_1), le2(t_1, t_2), bi1(t_2), re1(t_2), sq1(t_2)$
“the red square to the right of the blue circle”	$bl1(t_1), ci1(t_1), le2(t_2, t_1), re1(t_2), sq1(t_2)$
“the circle to the left of the square”	$ci1(t_1), le2(t_1, t_2), sq1(t_2)$

Table 1: Examples of linguistic utterances and their formal representations related to the situation $s = sm1(t_1), bl1(t_1), ci1(t_1), le2(t_1, t_2), bi1(t_2), re1(t_2), sq1(t_2)$.

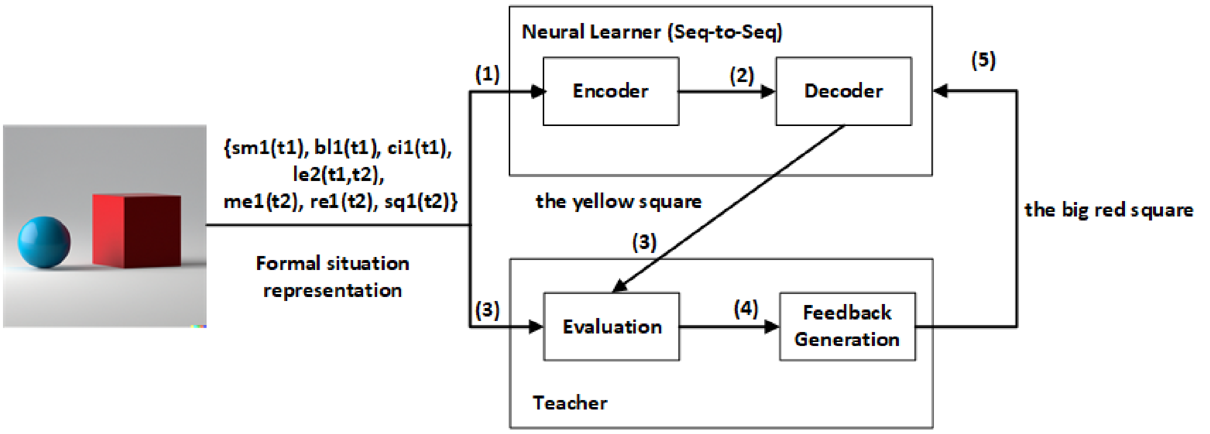


Figure 4: Interaction between learner and teacher models. (1) The situation’s formal representation is passed to the learner’s Encoder. The image shown here is for visualization purposes, each situation is written in formal representation. (2) The encoded situation is passed to the Decoder. (3) The generated utterance by the decoder, is passed to the Teacher’s evaluation module along with the situation. (4) The feedback generation module receives the situation representation if any error is detected. (5) The feedback generation module generates an utterance following the feedback strategies mentioned in Section 3.3 so the learner uses it for training as a means of correction.

valid utterance corresponding to the situation.

Evaluation module. This module evaluates if an utterance is part of the set of valid utterances corresponding to the situation. It uses the transducer T to classify an utterance as “syntax error”, “error in meaning”, or “correct”. An utterance is classified as “syntax error” when the transducer is unable to extract a corresponding meaning from the utterance, “error in meaning” when the meaning is extracted from the transducer correctly but the utterance is not part of the set of valid utterances for the current situation or “correct” otherwise.

Feedback Generation module. This module selects an utterance from the set of valid utterances from the situation to provide corrective feedback whenever an error is detected. This selection varies according to the feedback strategies mentioned below.

3.3 Feedback Strategies

For each situation, there are multiple semantically correct utterances that describe it. While the learner model generates an utterance, the teacher model faces the challenge of choosing an appropriate utterance to correct the learner, while the teacher is not aware of the learner’s intention (Lee et al., 2010). For this paper, we propose three feedback strategies that the teacher uses to select an utterance from the set of valid utterances to provide feedback: full-length, random, and lexical distance. We analyze the effect these strategies have on the learner’s performance. Each of these strategies shows different scenarios or outcomes of each interaction. Table 2 shows examples on what utterances the teacher provides according to each strategy.

Full-Length Feedback. This strategy presents an scenario where the teacher chooses an utterance that provides a complete description of the situation. A complete description contains two objects with all of its properties (i.e. size, color, and shape) and a relation between both objects. This allows a more efficient learning of the association between words and predicates present in the situation.

Random Feedback. This strategy presents a scenario where the teacher provides randomly selected corrections from a set of valid utterances corresponding to the situation. This strategy allows the learner model to be exposed to a wider distribution of possible utterances that describe the observed situation.

Lexical Distance Feedback. This strategy shows a targeted approach where the teacher identifies errors from the learner’s utterance by measuring the minimum edit distance or lexical distance between the learner utterance to some valid response, at a word-level, which it then chooses a correction that is grammatically different to the learner's utterance while preserving its intended meaning.

	Full-length FB	**Random FB**	**Lexical Distance FB**
Learner	the small red triangle to the right of the big yellow triangle	the big triangle	the small circle
Teacher	the small red triangle to the right of the big orange triangle	the triangle to the left of the small triangle	the small triangle

Table 2: Example of learner utterances and teacher utterances for the situation s = {bi1(t1), or1(t1) tr1(t1), le2(t1,t2), sm1(t2), re1(t2), tr1(t2)} *FB: Feedback.

Lexical Distance Feedback can be interpreted as a form of recast due to how it provides feedback while keeping the original meaning of the learner utterance. In terms of neural models, it helps to reduce larger penalty using cross-entropy loss. While Full-length feedback and Random Feedback can be interpreted as reformulations, these feedback strategies can provide utterances that could have a different meaning than the learner's utterance. With this, we explore whether providing these type of feedback affects the diversity of the learner's production of valid utterances. In other words, providing feedback that has a wider range of possible utterances like Random feedback allows the model to generate different valid utterances than target feedback like Lexical Distance Feedback which provides feedback with specific corrections.

3.4 Corrective Feedback Frequency

Another aspect that we are interested in exploring is how the frequency in which feedback is provided could affect the learner's utterance production and perception. To address this we propose two frequency approaches: (a) Corrective (CO) and (b) Non-Corrective (NC). Corrective feedback is the case which the teacher only provides a feedback utterance when the learner model generates an invalid utterance while Non-Corrective feedback is the case where the teacher always provides a feedback utterance regardless of the validity of the learner's utterance.

We want to observe whether the performance of the model could affirm the assumptions that providing feedback more frequently could lead to better performance. There are discussions that providing limited feedback on particular tasks is not sufficient to help an individual correct any observed errors, while providing too much feedback can overwhelm an individual and might lead them to make more errors. In our experiments, we want to observe whether providing corrective feedback only when the learner's utterance contained errors could lead to similar or better performance than providing non-corrective feedback. This could help generate training paradigms that focuses on addressing the frequency and quality of feedback which can reduce the amount of training data needed for language learning tasks.

4 Experimental Settings

All learner models were implemented using Pytorch (Paszke et al., 2019). The encoder and decoder GRUs have a hidden size of 300. The situation inputs are passed into a one-hot embedding layer forming a context vector representation. We used the Adam optimizer with a learning rate of 0.001. All learner models are trained with 18,000 situation-utterance set pairs and evaluated with 5,000 pairs. The learner's evaluation performance is recorded every 500 interactions.

4.1 Production setting

The goal of this setting is to observe how the feedback strategies and frequency of feedback affect the learner's utterance generation. We aim to address the following questions: (1) Can the learner provide a valid description of the situation? (2) Can the learner generate all possible utterances for any given situation? To address both of these questions, we propose two evaluation metrics: Semantic Accuracy and Completeness.

Semantic Accuracy measures the model's capacity to produce an utterance whose formal representation is accurate to the formal representation of the situation. In other words, we want the learner to generate utterances that preserve the meaning of the situation. We observe the model's utterance production development where it computes the loss with a single feedback utterance from the teacher model.

Completeness measures the rate of generated utterances that are present in the set of valid utterances. We want to see how many descriptions of a situation the learner model can learn during interactions with the teacher. The model uses a beam search approach to generate top-k utterances where $k = 40$ and the beam width to 22 (i.e. the vocabulary size). This technique has been applied to various tasks like story generation that require the generation of multiple sequences Fan et al. (2018); Holtzman et al. (2019).

4.2 Perception setting

For this setting, we evaluate the learner's capacity to choose utterances that correctly describe a given situation over other utterances that have at least one error in their description of the situation. Many psycholinguistic studies have used a two-alternative forced choice (2AFC) task to explore a child's preference for relevant objects for some linguistic stimuli (Gertner and Fisher, 2012; Bergelson and Swingley, 2012). This evaluation has been adapted to evaluate computational models for semantic evaluation of language models as the one shown in Nikolaus and Fourtassi (2021b).

Using the situation-utterance set pairs discussed in Section 2.3, we create triplet pairs $x_i = (s_i, u_t, u_d)$, where s_i is the given situation, u_t is the target utterance which is randomly selected from the set of valid utterance for that situation, and u_d is a distractor utterance that is similar to the target utterance, except that it has one instance of the evaluated concept replaced with another instance of the same concept.

For this task, we compute the probability of the target utterance given the situation and the distractor utterance given the situation. If the probability for the target utterance is greater than the probability of the distractor utterance, we consider that the learner model has successfully understood the evaluated concept. We train the models with the same 18,000 situation-utterance set pairs as the ones used in the production setting. For each concept, we create a set of 5,000 triplets for evaluation. We record the learner's performance after every 500 interactions.

5 Results

Syntactic Errors. Our model does not show any significant amount of syntax errors after a few interactions. Figure 5 shows the number of utterances that were classified as "syntax error" by the teacher's evaluation module for all models during the first 200 interactions. After 30 interactions, we see that corrective random feedback, non-corrective random feedback and non-corrective lexical distance have all of its generated utterances during the evaluation classified as "syntax error". Later interactions, we see that syntax errors become non-existent because the model is able to generate utterances that follow the expected grammar. Each learner model is initialized with the same random seed to ensure the comparison on the effects of each feedback strategy are from the same starting point. This high syntax error after 30 interactions present a point where the models were starting to get utterances with repeated shape concepts at different position before learning that size and color concepts occur before shape concepts, thus adjusting to the grammar.

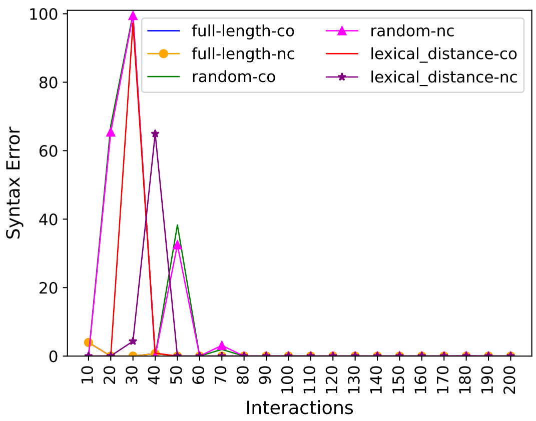


Figure 5: Syntax Errors for each feedback strategy with corrective and non-corrective feedback for the first 200 interactions. All models are evaluated after every 10 interactions with 5000 situation-utterance pairs.

Semantic Accuracy. Figure 6 shows the learner's semantic accuracy for each feedback strategy and feedback frequency. It is observed that corrective random feedback and corrective lexical distance has higher semantic accuracy than all other models while non-corrective and corrective full-length feedback have the lowest semantic accuracy. There were cases where the learner model described the situation using one relation while the teacher provided a valid feedback with the opposite relation. This causes confusion to the model due to misaligned objects. For example, the learner generated the utterance "the big red star to the left of the small yellow circle" and the teacher provided "the small yellow circle to the right of the big red star". Corrective random feedback and lexical distance shows oscillations between interactions. We observed that these feedback strategies provide utterances of different lengths more frequent than full-length feedback which causes the learner model to fail to describe some situations.

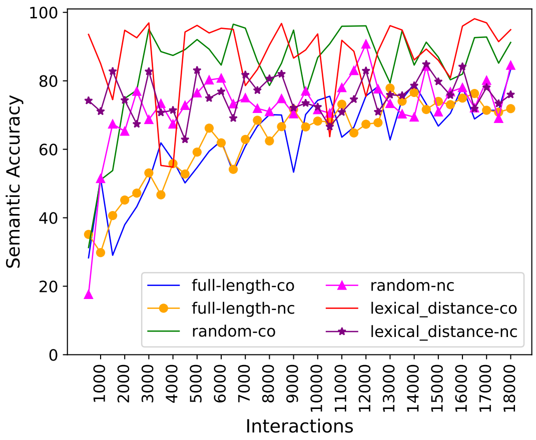


Figure 6: Semantic Accuracy evaluation for each feedback strategy with corrective and non-corrective feedback.

As shown in Table 3 most models reach an accuracy of sixty percent or more within the first 500 interactions for lexical distance, 1000 interactions for random feedback, and 3000 interactions for full-length feedback. Random feedback and lexical distance are the only strategies that reach ninety percent semantic accuracy. Corrective lexical distance feedback reaches ninety percent after 500 interactions.

The learner trained with lexical distance feedback strategy correctly generate valid utterances earlier than other strategies because these generated utterances only describe the shape of at least one object in the situation. On the other hand, full-length feedback expects the learner model to describe the size, color, and shape of both objects which requires more interactions for the learner to successfully describe a situation following this strategy.

All feedback strategies with corrective frequency are able to reach each performance threshold with less interaction than non-corrective frequency. This is interesting because it shows that for this task, limiting the amount of corrective feedback helps the learner generate more valid utterances for each situation.

learner \ accuracy	**60%**	**70%**	**80%**	**90%**
full-length-co	3000-3500	7500-8000	17500-18000	-
full-length-nc	5000-5500	8500-9000	-	-
random-co	1500-2000	1500-2000	2500-3000	2500-3000
random-nc	1000-1500	2000-2500	5000-5500	11500-12000
lexical-distance-co	0-500	0-500	0-500	0-500
lexical-distance-nc	0-500	0-500	1000-1500	-

Table 3: Number of interactions where the learner model reached or exceeded certain semantic accuracy threshold percentage. Since each model is evaluated every 500 interactions, we show the interval range where the model reached the specified accuracy percentage.

Completeness. Figure 7 shows the learner completeness score at various interactions between the models. It shows that non-corrective feedback strategies have higher scores than corrective feedback strategies. Non-corrective random feedback has the highest completeness score at 55%, which means that the learner model trained with this strategy and frequency was able to generate around 20 different valid utterances. On the other hand, corrective full-length feedback only generated 3 valid utterances which is approximately 7%.

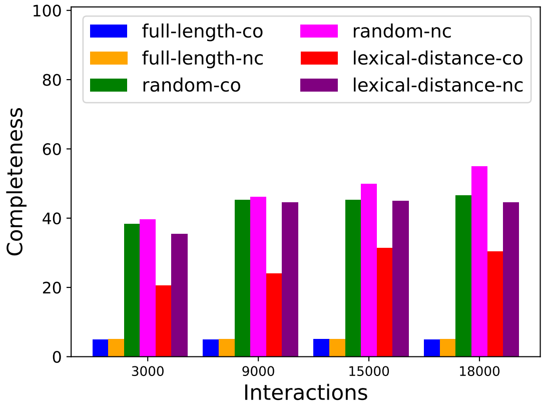


Figure 7: Completeness evaluation for each feedback strategy with corrective and non-corrective feedback.

When comparing the completeness score and the semantic accuracy, there is no indication that high semantic accuracy also means high completeness. Instead, we observed that full-length feedback had low semantic accuracy and low completeness score. Also, both random feedback and lexical distance feedback had high semantic accuracy and high completeness. A possible explanation for this lies on the diversity of feedback utterances the learner receives. Random feedback provides utterances of different lengths whereas full-length feedback only provides one or two utterances with the full description of the situation. We can also argue that Lexical distance feedback is in the middle in terms of performance because this strategy selected utterances with minimal edit-distance to the learner's utterance, thus limiting the range of possible utterances. We believe that exposing the learner to different utterances are helpful for the learner model to develop its set of utterances to describe each situation.

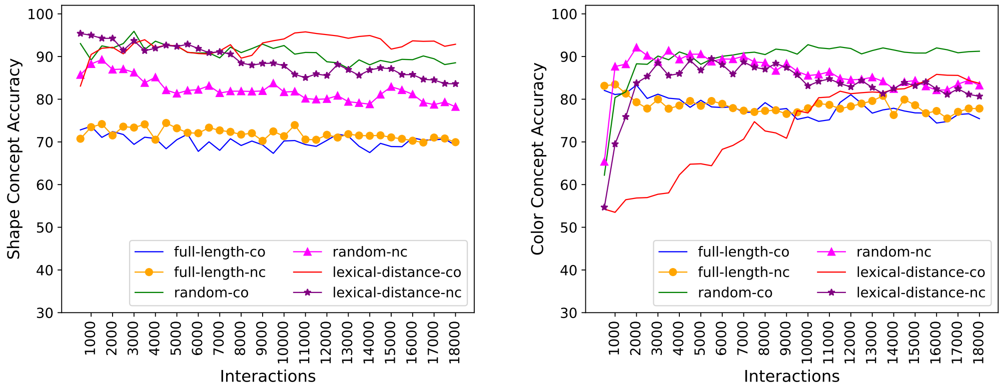
Perception Evaluation.

Figure 8 show the perception evaluation for every concept with each feedback strategy and feedback frequency. We observe that there is no significant difference between the corrective and non-corrective frequency of each feedback strategy. Corrective full-length feedback has the lowest shape accuracy and highest size accuracy. We observe that corrective lexical distance has low accuracy for color, size, and relation concept during early interactions but increases its accuracy to be on par with other feedback strategies.

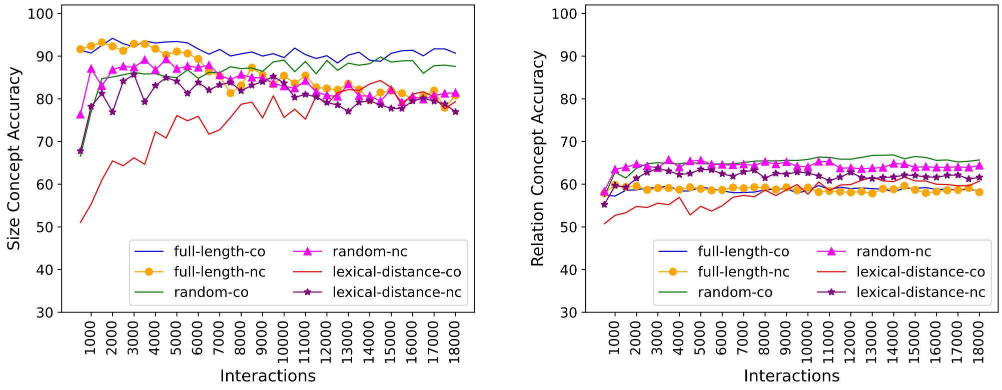
All learner models are able to have concept accuracy above 50%, therefore we could argue that these models are able to learn these concepts. There are some cases where some models have difficulty learning certain concepts. One case is the relation concept where all models have low accuracy. For example, full-length feedback strategy, the learner has learned to generate an utterance like "the big red star to the left of the small orange circle" but the current concept evaluation test expects a target utterance like "the small orange circle to the right of the big red star". These utterances while both equally valid, the learner model might not understand the target utterance due to the positioning of the objects. For random and lexical feedback, the learner did not learn to generate utterances with relations.

6 Discussion

In the production setting, each feedback strategy presents forms in which a teacher could provide feedback to a learner as guided responses, from describing the full situation to rephrasing the utterance to correct incorrect or missing concepts. The learner models had a positive response to these guided types of responses which yielded higher semantic accuracy. However, we noticed that each model generated utterances of specific concept combinations. This is also supported by the completeness evaluation which shows the set of utterances the model is able to associate to each situation. Random feedback strategy generated utterances of different length for each situation. These results indicate that exposing the learner model to various feedback utterances across multiple situations can improve the model's semantic accuracy as well as completeness score.



(a) Shape Concept (b) Color Concept



(c) Size Concept (d) Relation Concept

Figure 8: Perception Task evaluation for all feedback strategies within the corrective and non-corrective setting.

In the perception setting, we measure a model's concept accuracy which indirectly shows how the model develops word-meaning associations when the model successfully selects a target utterance which has the correct use of a concept as well as correctly describing the current situation. Each feedback strategy present a constant concept accuracy throughout the interactions.

We analyzed aggregated attention maps during the model's production setting to measure the word-meaning mappings were developed as new interactions occurred. The maps for full-length feedback show that this strategy help the learner model have strong one-to-one association between concept words and their respective predicates. For random feedback and lexical distance feedback strategies, the models formed associations between each word to multiple predicates corresponding to different concepts. Further details and attention maps can be found in Appendix A

In addition to analyze attention maps, we observed the error rates of every concept for each feedback strategy. Overall, the error rates for each concept decreases as the learner interacts with the teacher. Some strategies like random and lexical distance feedback have higher error rates at earlier interactions, they still decrease during later interactions. Additional details are in Appendix B

Although the evaluation metrics in our experiments aim to explore the acquisition of a natural language by neural models, they are not sufficient and thus more complex metrics are required. For a fair comparison between neural language acquisition and human language acquisition, establishing cognitive plausible neural architectures. By cognitive plausible, we mean that neural models could emulate human-like processing (Beinborn and Hollenstein, 2023). We believe that this framework has potential to include additional metrics that can evaluate neural model learning and gain insights to language acquisition theories.

7 Conclusion

This work proposes a framework to explore teacher-learner interactions with corrective feedback within a controlled environment using formal representations of objects and their properties. We evaluate various feedback strategies and their influence on the learner model's utterance generation for a given situation and perception of different concepts like shape, color, size, and relation present in a situation. These results show that the learner models can generate a different subset of valid utterances to describe a situation according to the feedback strategy employed by the teacher. Some strategies like random and lexical distance were useful for the learner model to learn multiple utterances whereas full-length only allowed the learner model to learn 1-2 valid utterances. In terms of perception, some models present challenges for certain concepts due to unseen target utterances during concept evaluation that were not provided by the teacher during training interactions.

Given the reduced size of the vocabulary and the concepts, the use of GRU was selected to better highlight the main idea of creating a controlled framework that could be insightful, in terms of production and perception of a synthetic language, for the community to study the challenges of language acquisition and in-depth semantic evaluation of generated utterances. Further study and technical developments are needed to possibly incorporate newer neural models into this framework. In this paper, the teacher's feedback served as the ground truth labels used by the learner model. The learner model did not do any analysis or additional processing of the teacher's feedback which might limit the effectiveness of feedback and interactions For future work, we need to design procedures where the learner model can interpret the teacher's feedback and use said interpretation in the production process. One approach is to introduce mechanisms that allows the learner model to use the formal representation of the teacher's utterance to incorporate semantics into the loss function calculations.

8 Limitations

The framework has some limitations that need to be addressed. First, this framework uses synthetic data within a controlled environment. Natural language is very complex, which makes it a challenging task to create evaluation frameworks. Our data do not fully represent a natural language. Second, the proposed models to evaluate the data are basic in design. Our models rely on single-layer GRU with sequential formal representations of situations. We need to explore the use of this data on other models to have a wider panel of performance for better comparison and evaluation. Although our intent is to study child language acquisition, our work does not have any empirical analysis between the neural models and other studies involving actual children. Our goal is to explore additional metrics and evaluation settings in which this framework compares neural learner models with human performance.

References

Dana Angluin and Leonor Becerra-Bonache. 2017. A model of language learning with semantics and meaning-preserving corrections. *Artificial Intelligence*, 242:23 – 51.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate. *Preprint*, arXiv:1409.0473.

Marco Baroni. 2022. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. *Preprint*, arXiv:2106.08694.

L. Beinborn and N. Hollenstein. 2023. *Cognitive Plausibility in Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Springer International Publishing.

Tony Belpaeme and Anthony Morse. 2012. Word and category learning in a continuous semantic domain: Comparing cross-situational and interactive learning. *Advances in Complex Systems*, 15(03n04):1250031.

Elika Bergelson and Daniel Swingley. 2012. At 6–9 months, human infants know the meanings of many common nouns. *Proceedings of the National Academy of Sciences*, 109(9):3253–3258.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michelle M. Chouinard and Eve V. Clark. 2003. Adult reformulations of child errors as negative evidence. *Journal of Child Language*, 30(3):637–669.

Herbert H. Clark. 1996. *Using Language*. 'Using' Linguistic Books. Cambridge University Press.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint*.

Afsaneh Fazly, Afra Alishahi, and Suzanne Stevenson. 2010. A probabilistic computational model of cross-situational word learning. *Cognitive Science*, 34(6):1017 – 1063.

Jerome A Feldman, George Lakoff, Andreas Stolcke, and Susan Hollbach Weber. 1990. Miniature language acquisition: A touchstone for cognitive science. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pages 686–693. Citeseer.

Stefan L. Frank, Padraic Monaghan, and Chara Tsoukala. 2019. Neural network models of language acquisition and processing. In *Human Language: From Genes and Brains to Behavior*. The MIT Press.

Yael Gertner and Cynthia Fisher. 2012. Predicted errors in children's early sentence comprehension. *Cognition*, 124(1):85–94.

Lila Gleitman. 1990. The structural sources of verb meanings. *Language Acquisition*, 1(1):3–55.

Sarah Hiller. 2016. Corrective feedback in first language acquisition.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Alexis Juven and Xavier Hinaut. 2020. Cross-situational learning with reservoir computing for language acquisition modelling. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

S.D. Krashen, S.D. Krashen, and T.D. Terrell. 1983. *The Natural Approach: Language Acquisition in the Classroom*. Language Teaching Methodology Series. Pergamon Press.

Patricia K. Kuhl, Feng-Ming Tsao, and Huei-Mei Liu. 2003. Foreign-language experience in infancy: Effects of short-term exposure and social interaction on phonetic learning. *Proceedings of the National Academy of Sciences*, 100(15):9096–9101.

Sungjin Lee, Cheongjae Lee, Jonghoon Lee, Hyungjong Noh, and Gary Geunbae Lee. 2010. Intention-based corrective feedback generation using context-aware model. In *CSEDU*.

Michael H. Long. 1981. Input, interaction, and second-language acquisition. *Annals of the New York Academy of Sciences*, 379(1):259–278.

Roy Lyster and Leila Ranta. 1997. Corrective feedback and learner uptake: Negotiation of form in communicative classrooms. *Studies in Second Language Acquisition*, 19(1):37–66.

Padraic Monaghan, Simón Ruiz, and Patrick Rebuschat. 2021. The role of feedback and instruction on the cross-situational learning of vocabulary and morphosyntax: Mixed effects models reveal local and global effects on acquisition. *Second Language Research*, 37(2):261–289.

Mitja Nikolaus and Abdellah Fourtassi. 2021a. Evaluating the acquisition of semantic knowledge from cross-situational learning in artificial neural networks. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 200–210, Online. Association for Computational Linguistics.

Mitja Nikolaus and Abdellah Fourtassi. 2021b. Modeling the interaction between perception-based and production-based learning in children's early acquisition of semantic knowledge. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 391–407, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Steven Pinker. 2013. *Learnability and Cognition: The Acquisition of Argument Structure (1989/2013)*, new edition edition. Cambridge, MA: MIT Press.

Eva Portelance and Masoud Jasbi. 2023. The roles of neural networks in language acquisition. *Retrieved from osf. io/preprints/psyarxiv/b6978*.

W. V. O. Quine. 1960. *Word & Object*. MIT Press.

J. Sachs, B. Bard, and M. Johnson. 1981. Language learning with restricted input: Case studies of two hearing children of deaf parents. *Applied Psycholinguistics*, 2:33–54.

Ted Schoneberger. 2010. Three myths from the language acquisition literature. *The Analysis of Verbal Behavior*, 26:107–131.

Kenny Smith, Andrew DM Smith, and Richard A Blythe. 2011. Cross-situational learning: An experimental study of word-learning mechanisms. *Cognitive Science*, 35(3):480–498.

Linda Smith and Chen Yu. 2008. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106:1558–68.

Luc Steels. 1995. A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332.

Sumarga Suanda, Nassali Mugwanya, and Laura Namy. 2014. Cross-situational statistical word learning in young children. *Journal of experimental child psychology*, 126.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Alex Warstadt and Samuel R. Bowman. 2024. What artificial neural networks can tell us about human language acquisition. *Preprint*, arXiv:2208.07998.

Chen Yu and Dana H Ballard. 2004. A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perception (TAP)*, 1(1):57–80.

Chen Yu and Linda B. Smith. 2007. Rapid word learning under uncertainty via cross-situational statistics. *Psychological Science*, 18:414 – 420.

A Word Learning Evaluation

Specifically, it shows the attention maps for each feedback strategy and feedback frequency after 18,000 interactions.

The learner models trained with corrective full-length feedback has strong word-predicate associations for all concepts. For example, the model trained with corrective full-length feedback shows that the word “hexagon” has a strong association with the predicate “he1” while having little to no association to any other predicates. However, models trained with non-corrective full-length feedback show weaker associations between words and predicates in the sense that one word is associated to more than one predicate. For example, the word “big” is associated to the predicates “bi1”as well as other color predicates like “ye1”, “gr1”, and “re1”. Also, these models showed strong associations for the relation concept words “above” and “below” to predicate “ab2” as defined in the grammar discussed in Section 2.2. Unfortunately, the model did not present any strong associations between the words “left” and “right” to the predicate “le2”. This could occur because the model was not provided with sufficent feedback utterances that uses both of these relations.

Models trained with corrective random feedback, as shown in Figure 9b, have strong associations for the shape concepts. However, for color, size, and relations concepts, we see that the model has higher attention towards shape and relation predicates. For example, the word “red” has associations to color predicates like “st1”, “ci1” and even relation predicates “ab2” and “le2”. Possibly this model receives feedback utterances that might contain instances of color and size concepts, which are not required for an utterance to be valid, at different positions thus making it difficult to fully adjust the attention weights. In Figure 9e shows that non-corrective random feedback present the same issue.

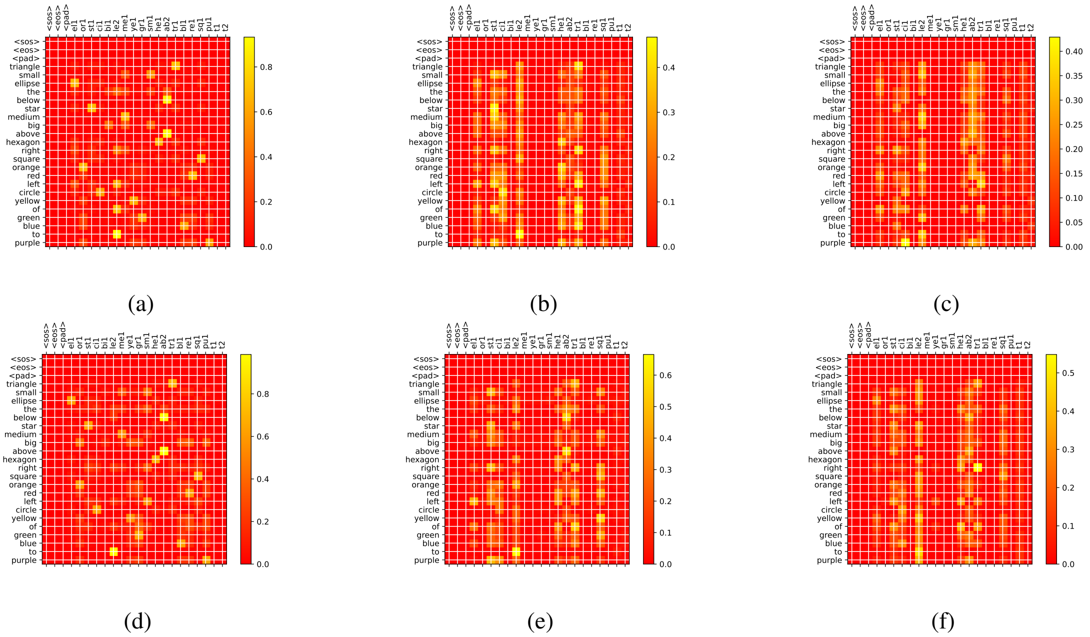
Models trained with lexical distance feedback shows that most words had strong association to relation predicates. For example, the word “triangle” has high association to the predicates “le2” and “ab2”. Also, we observe that words of shape instances have high association to its corresponding predicate. Similarly to random feedback, this strategy also has difficulty in properly update the attention weights for different concepts. The model associates the words for sizes like “small”, “medium”, and “big” to various shape predicates. The same is observed for color concept which also associate each color word to the shape predicates.

We can say that these attention maps are consistent with the behavior of each feedback strategy. Full-length feedback provide the full description of the situation which makes the attention between words and predicates simple to compute. Random and lexical distance feedback provide utterances with partial descriptions which can affect how certain words are aligned to the situation. For example, the utterance “the red circle” being aligned to the situation “bi1(x1) re1(x1) ci1(x1) le2(x1,x2) sm1(x2) or1(x2) tr1(x2)” might align the word “red” to the predicate “bi1” which occurs before the predicate “re1”.

B Concept Error Rate Analysis.

We conducted an analysis of the frequency of concept errors during the production evaluation. Table 4 shows the error rates (i.e., the number of concept errors divided by the number of examples presented during evaluation) for each concept across different interaction periods. The relation concept had values lower than 1% therefore were not included in this table. These results indicate that the error rate trends depend on the feedback strategy employed as explained below.

Strategies with corrective feedback have a lower error rate than non-corrective feedback across all strategies for all concepts. Also, we observe that for shape and size concepts, full-length feedback has a decreasing error rate as it interacts with the teacher model. However this strategy has higher error rates for color concepts. We hypothesize that the model generate ut-



(a) (b) (c)

(d) (e) (f)

Figure 9: Aggregated attention maps for each feedback strategy and feedback frequency after all 18000 interactions. This values are the average attention weights for 5000 evaluation interactions. Attention maps for feedback strategies with corrective feedback, (a) Full-length feedback, (b) Random feedback, (c) Lexical Distance feedback. Attention maps for feedback strategies with non-corrective feedback, (d) Full-length feedback, (e) Random feedback, (f) Lexical Distance feedback.

terances with shape instances, followed by utterances with size and shape instances before learning color instances. Also, we have cases where the learner receives teacher feedback with the opposite relation from what the learner generated. This causes the learner to learn incorrect color instances for the intended object. Another thing to notice is that for random feedbaqck and lexical distance feedback strategies, the error rates tend to be lower than full-length feedback for early interactions while presenting higher error rates at later interactions. This occurs because both of these strategies are trained with partial descriptions of a situation, unlike full-length feedback, making them less prone to generate incorrect utterances.

For the relation concept, we observe that no errors were detected due to the model either not generating utterances that do not have any instances of relations or the relation were correctly generated. For example, a given situation could refer to a big star that is above a small yellow square as "the big red star above the small yellow square" rather than "the small yellow square below the big red star". These two sentences are both valid, however, their relation concepts refers to the objects in opposite perspectives.

This shows how these models are able to minimize the error rate in terms of utterance generation with the corresponding concepts as the interactions with the teacher increases. Although some models have a large error rate at the early interactions, we believe that these errors occur due to the utterances provided by the teacher model which could be valid to describe the given situation, but might not be fully aligned to what the learner model intended to describe. An example of this can be seen in the case of the color concept for the corrective full-length feedback strategy where the learner model attempts to describe "the big yellow circle" but the teacher model provides "the small blue star above the medium yellow circle" which by our CE-loss, it will train the learner to better learn to describe "the small blue star".

<table>
<tr><td colspan="11">Shape Concept</td></tr>
<tr><td></td><td><b>500</b></td><td><b>2500</b></td><td><b>4500</b></td><td><b>6500</b></td><td><b>8500</b></td><td><b>10500</b></td><td><b>12500</b></td><td><b>14500</b></td><td><b>16500</b></td><td><b>18000</b></td></tr>
<tr><td><b>full-length-co</b></td><td>14.29</td><td>3.28</td><td>2.91</td><td>2.76</td><td>3.87</td><td>0.12</td><td>2.54</td><td>1.03</td><td>0.45</td><td>1.35</td></tr>
<tr><td><b>full-length-nc</b></td><td>11.87</td><td>3.11</td><td>1.01</td><td>3.18</td><td>1.93</td><td>1.64</td><td>1.15</td><td>1.83</td><td>3.14</td><td>1.87</td></tr>
<tr><td><b>random-co</b></td><td>15.00</td><td>9.37</td><td>1.65</td><td>7.51</td><td>1.21</td><td>0.76</td><td>1.88</td><td>2.93</td><td>0.75</td><td>0.34</td></tr>
<tr><td><b>random-nc</b></td><td>15.64</td><td>6.68</td><td>3.68</td><td>2.54</td><td>0.85</td><td>1.97</td><td>2.84</td><td>3.18</td><td>3.03</td><td>2.43</td></tr>
<tr><td><b>lexical-distance-co</b></td><td>15.22</td><td>8.15</td><td>12.11</td><td>8.27</td><td>14.20</td><td>2.62</td><td>6.87</td><td>6.30</td><td>8.87</td><td>6.82</td></tr>
<tr><td><b>lexical-distance-nc</b></td><td>11.06</td><td>9.55</td><td>12.70</td><td>9.95</td><td>5.39</td><td>6.73</td><td>4.11</td><td>8.08</td><td>9.49</td><td>7.60</td></tr>
<tr><td colspan="11">Color Concept</td></tr>
<tr><td></td><td><b>500</b></td><td><b>2500</b></td><td><b>4500</b></td><td><b>6500</b></td><td><b>8500</b></td><td><b>10500</b></td><td><b>12500</b></td><td><b>14500</b></td><td><b>16500</b></td><td><b>18000</b></td></tr>
<tr><td><b>full-length-co</b></td><td>14.33</td><td>17.20</td><td>18.37</td><td>19.91</td><td>21.65</td><td>23.94</td><td>22.55</td><td>23.43</td><td>23.99</td><td>24.22</td></tr>
<tr><td><b>full-length-nc</b></td><td>12.83</td><td>18.97</td><td>21.36</td><td>21.14</td><td>22.81</td><td>22.62</td><td>23.23</td><td>23.36</td><td>22.57</td><td>23.28</td></tr>
<tr><td><b>random-co</b></td><td>10.07</td><td>3.89</td><td>5.88</td><td>13.01</td><td>5.59</td><td>4.19</td><td>6.97</td><td>9.83</td><td>2.24</td><td>7.69</td></tr>
<tr><td><b>random-nc</b></td><td>15.70</td><td>6.59</td><td>10.47</td><td>8.98</td><td>4.60</td><td>11.50</td><td>14.16</td><td>6.45</td><td>9.43</td><td>5.82</td></tr>
<tr><td><b>lexical-distance-co</b></td><td>7.53</td><td>3.44</td><td>0.17</td><td>1.92</td><td>8.80</td><td>11.80</td><td>6.47</td><td>12.78</td><td>5.65</td><td>1.48</td></tr>
<tr><td><b>lexical-distance-nc</b></td><td>13.40</td><td>6.62</td><td>5.94</td><td>5.16</td><td>13.72</td><td>9.12</td><td>11.78</td><td>5.21</td><td>6.45</td><td>10.46</td></tr>
<tr><td colspan="11">Size Concept</td></tr>
<tr><td></td><td><b>500</b></td><td><b>2500</b></td><td><b>4500</b></td><td><b>6500</b></td><td><b>8500</b></td><td><b>10500</b></td><td><b>12500</b></td><td><b>14500</b></td><td><b>16500</b></td><td><b>18000</b></td></tr>
<tr><td><b>full-length-co</b></td><td>11.09</td><td>10.85</td><td>9.26</td><td>7.65</td><td>8.32</td><td>4.04</td><td>4.00</td><td>2.37</td><td>1.86</td><td>1.35</td></tr>
<tr><td><b>full-length-nc</b></td><td>12.07</td><td>8.66</td><td>6.42</td><td>5.23</td><td>2.41</td><td>4.10</td><td>3.97</td><td>2.27</td><td>3.29</td><td>2.81</td></tr>
<tr><td><b>random-co</b></td><td>8.92</td><td>14.16</td><td>18.66</td><td>6.07</td><td>20.05</td><td>20.97</td><td>18.03</td><td>13.51</td><td>20.53</td><td>17.48</td></tr>
<tr><td><b>random-nc</b></td><td>10.72</td><td>15.87</td><td>14.02</td><td>16.39</td><td>21.13</td><td>13.93</td><td>10.03</td><td>16.90</td><td>13.65</td><td>17.76</td></tr>
<tr><td><b>lexical-distance-co</b></td><td>3.18</td><td>6.87</td><td>4.18</td><td>7.66</td><td>3.86</td><td>13.97</td><td>12.10</td><td>6.02</td><td>11.02</td><td>16.01</td></tr>
<tr><td><b>lexical-distance-nc</b></td><td>4.89</td><td>9.89</td><td>10.29</td><td>10.92</td><td>8.36</td><td>10.13</td><td>9.68</td><td>11.94</td><td>9.54</td><td>7.50</td></tr>
</table>

Table 4: Error rates for each concept after training interactions. Measures the number of learner utterances classified as error in meaning due to the incorrect use of one of the concepts.

Relation-Aware Prompting Makes Large Language Models Effective Zero-shot Relation Extractors

Mahdi Rahimi, Razvan-Gabriel Dumitru, Mihai Surdeanu
Department of Computer Science
University of Arizona, Tucson, Arizona, USA
`{marahimi, rdumitru, msurdeanu}@arizona.edu`

Abstract

While supervised relation extraction (RE) models have considerably advanced the state-of-the-art, they often perform poorly in low-resource settings. Zero-shot RE is vital when annotations are not available either due to costs or time constraints. As a result, zero-shot RE has garnered interest in the research community. With the advent of large language models (LLMs) many approaches have been proposed for prompting LLMs for RE, but these methods often either rely on an accompanying small language model (e.g., for finetuning on synthetic data generated by LLMs) or require complex post-prompt processing. In this paper, we propose an effective prompt-based method that does not require any additional resources. Instead, we use an LLM to perform a two-step process. In the first step, we perform a targeted summarization of the text with respect to the underlying relation, reduce the applicable label space, and synthesize examples. Then, we combine the products of these processes with other elements into a final prompt. We evaluate our approach with various LLMs on four real-world RE datasets. Our evaluation shows that our method outperforms the previous state-of-the-art zero-shot methods by a large margin. This work can also be considered as a new strong baseline for zero-shot RE that is compatible with any LLM[1].

1 Introduction

Relation extraction (RE) aims to identify semantic relations between two entities from unstructured text. With the recent advances in large language models (LLMs), studies show that LLMs perform well in various downstream tasks without any training or fine-tuning. But it is unclear whether they are effective for zero-shot RE. A recent line of research shows that such zero-shot approaches for relation extraction are ineffective or continue to lag behind supervised methods (Ma et al., 2023, Wang et al., 2023, Ye et al., 2023, Jimenez Gutierrez et al., 2022, Li et al., 2023a, Xu et al., 2023b, Han et al., 2024, Swarup et al., 2025). However, other line of LLM-based research reports results comparable or outperforming state-of-the-art. These methods fall into three groups. In a first group, these methods use LLMs to some extent, but eventually rely on fine-tuning a small language model (Xu et al., 2023b, Zhou et al., 2024, Xu et al., 2023a, Tang et al., 2023). For instance, an LLM is used to generate/augment synthetic data and then a model from BERT family is fine-tuned on the generated data. The second group fine-tunes a large language model (Wadhwa et al., 2023, Sainz et al., 2024, Li et al., 2024, Wang et al., 2023). Fine-tuning LLMs requires specialized hardware, significant compute resources, and is expensive. The third group does not require fine-tuning but requires complex post-prompt computations, e.g., Li et al. (2023c) performs a complex computation on the LLM answers using an uncertainty-based active learning method to estimate output probabilities of the LLM. A few other methods, e.g., (Wei et al., 2024, inter alia), that do not belong to the above groups are evaluated on limited benchmarks. Therefore, it is unclear whether prompt based zero-shot RE is effective without any finetuning or complex post-prompt computations.[2]

In this work, we present an effective prompt-based method to RE that does not require either fine-tuning or complex computations. Our approach only requires an API access to an LLM. This simplifies the zero-shot RE process and makes it more accessible and faster to deploy which is important for developing zero-shot systems. We achieve this by a novel prompt-based method we

---

[1] Code and data are available at `https://github.com/mahrahimi1/relation-aware-prompting`

[2] For a detailed review of the literature on zero-shot relation extraction see Appendix A.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 280–292
November 8-9, 2025 ©2025 Association for Computational Linguistics

call Relation-Aware Prompting. Formally:

(1) We perform a targeted summarization of instances with respect to the underlying relations to bring out the relations in the texts and discard unrelated facts.

(2) We reduce the applicable relation labels using annotation guidelines and through a method inspired by the process of elimination. We use entity type constraints for this purpose when they are available as well.

(3) We propose a method using subject-verb-object (SVO) structure to generate synthetic examples that will be used as demonstrations.

(4) We combine the results of the above processes with other elements such as relation definitions into a final prompt.

We evaluate our approach using various LLMs on four real-world and challenging relation extraction datasets. The evaluation shows that our method outperforms the previous state-of-the-art zero-shot methods by a large margin. We also perform an ablation study where we investigate the effectiveness and usefulness of our prompt elements that will demonstrate the effectiveness of the proposed method.

This work can also be considered as a new strong baseline for relation extraction. Any LLM-based work in RE (such as finetuning LLMs, or other methods) can use our method as a strong baseline for evaluating their respective approach.

2 Problem Statement

In the RE task, the goal is to classify a sentence containing two marked entities (a *head* and a *tail*) into a set of predefined relations, or determine that none of the relations apply (referred to as none-of-the-above or NoTA). This work focuses on zero-shot RE where no RE training data is provided to models prior to inference time.

3 Methodology

Our method is a two-step process. In the first step, (a) relations between the head and tail entities in instances are summarized; (b) the applicable label space is reduced; and (c) examples are synthesized. In the second step, the results of the first step are combined with other prompting elements into a final comprehensive prompt. The following subsections describe each step. Figure 1 demonstrates an overview of our approach.

3.1 Targeted Summarization

We summarize the relation between the head and tail entities in instances (Li et al., 2023c) in order to bring out the relation in the text and discard unrelated facts and misleading cues. The goal is not to summarize the complete sentence, but to summarize the relation between the entities in the sentence. In the prompt, we emphasize this and instruct the model to ignore everything else for the summary. A concrete example as well as our prompt is provided in Appendix C.

3.2 Reducing the Label Space

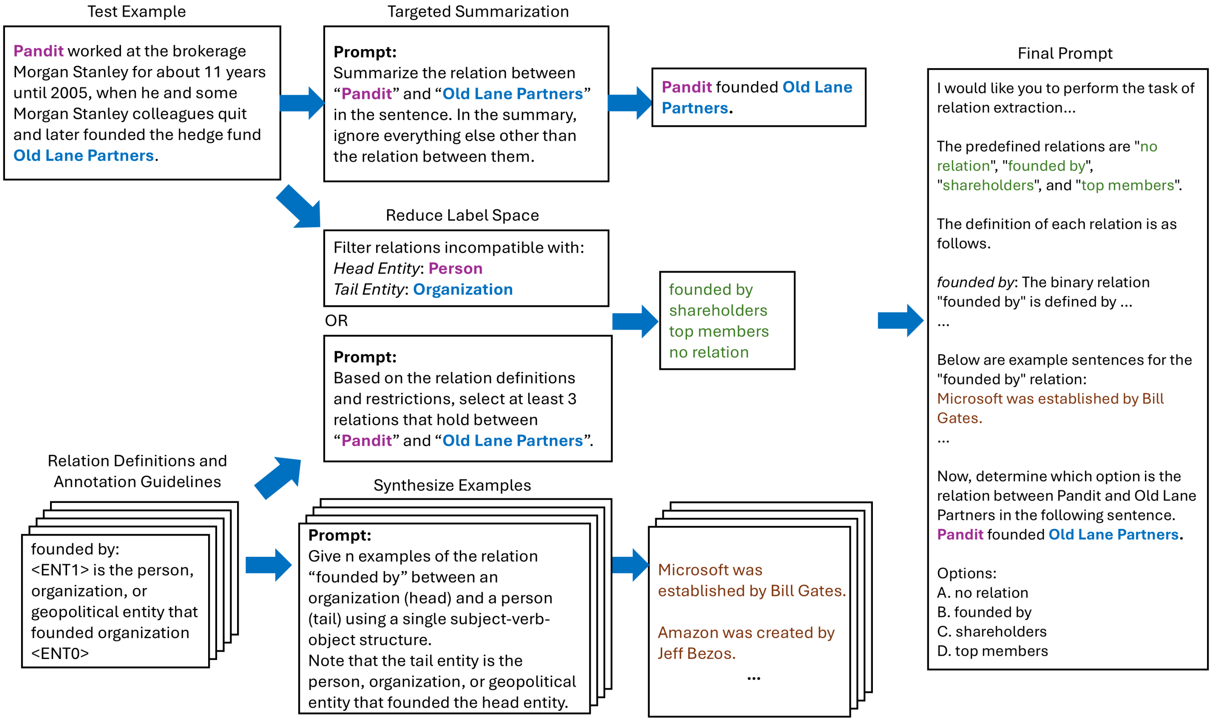
Relation Extraction usually involves classification between many classes. This is an overly difficult task for LLMs. When entity types are available in the data, we use them to filter out the relation types that are impossible. In case entity typing is unavailable or not applicable,[3] we propose an approach to reduce the number of candidate relations through a method inspired by the process of elimination. As the first step, we ask an LLM to reduce the number of classes down to 3 for each instance given the relation definitions and annotation guidelines. For example, Figure 4 (left) in Appendix F shows our prompt for SemEval 2010 Task 8 dataset. We select the parts of annotation guidelines that we believe are helpful for the LLM to differentiate between relation types given the entities. For instance, guidelines may have a "Restrictions" section in relation definitions that can help the LLM narrow down candidate relations based on entities of the test examples. Figure 4 (right) in Appendix F shows what we selected for Instrument-Agency relation in the aforementioned dataset.

After the candidate relations are narrowed down to three, we add NoTA (if not already included), and then prompt the LLM to select the best option as explained in subsection 3.4 and shown in Figure 6 (bottom) in Appendix F.

If the relations are undirected, i.e., it is not provided which entity in the sentence is the head and which entity is the tail, such as in SemEval 2010 Task 8 dataset, one extra step is required to determine the direction of the relation. Further details are provided in Appendix D.

---

[3] Such as SemEval 2010 Task 8 dataset where entities are not named entities, but rather common nouns.



Pandit worked at the brokerage Morgan Stanley for about 11 years until 2005, when he and some Morgan Stanley colleagues quit and later founded the hedge fund Old Lane Partners.

Prompt:
Summarize the relation between “Pandit” and “Old Lane Partners” in the sentence. In the summary, ignore everything else other than the relation between them.

Pandit founded Old Lane Partners.

Filter relations incompatible with:
Head Entity: Person
Tail Entity: Organization

Prompt:
Based on the relation definitions and restrictions, select at least 3 relations that hold between “Pandit” and “Old Lane Partners”.

founded by
shareholders
top members
no relation

founded by:
<ENT1> is the person, organization, or geopolitical entity that founded organization <ENT0>

Prompt:
Give n examples of the relation “founded by” between an organization (head) and a person (tail) using a single subject-verb-object structure.
Note that the tail entity is the person, organization, or geopolitical entity that founded the head entity.

Microsoft was established by Bill Gates.

Amazon was created by Jeff Bezos.

...

I would like you to perform the task of relation extraction...

The predefined relations are "no relation", "founded by", "shareholders", and "top members".

The definition of each relation is as follows.

founded by: The binary relation "founded by" is defined by ...
...

Below are example sentences for the "founded by" relation:
Microsoft was established by Bill Gates.
...

Now, determine which option is the relation between Pandit and Old Lane Partners in the following sentence.
Pandit founded Old Lane Partners.

Options:
A. no relation
B. founded by
C. shareholders
D. top members

Figure 1: Overview of Relation-Aware Prompting.

3.3 Synthesizing Examples

Some annotation guidelines have examples for each relation type, but we do not use them in our prompt to emulate a no-supervision scenario. Instead, we *synthesise examples*: we prompt an LLM to generate examples using a subject-verb-object (SVO) structure. We generate the examples based on relation definitions, entity types, and relation labels. Figure 5 in Appendix F shows our prompt. After generating the examples, we use them in our final prompt as demonstrations similar to in-context learning (explained in the next subsection).

3.4 Final Prompt

The results of previous processes, i.e, “the targeted summary”, “the reduced applicable labels”, and “synthesized examples” are combined with other prompting elements to form our final prompt shown in Figure 6 (bottom) in Appendix F. These other prompting elements are entity tagging (Zhou et al., 2024) and relation definitions and annotation guidelines (Zhou et al., 2024). Furthermore, we pose the final classification as multiple-choice question answering (Zhang et al., 2023a) where options are relation labels. Additionally, we turn the labels into a more human-readable form before using them as the options. For example, we change “org:founded_by” to “founded by”.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate our method on four relation extraction datasets: TACRED (Zhang et al., 2017), TACREV (Alt et al., 2020), RETACRED (Stoica et al., 2021), and SemEval-2010 Task 8 (Hendrickx et al., 2010) (henceforth SemEval). The statistics of the datasets are provided in Appendix B. We follow previous work (Sainz et al., 2021, Lu et al., 2022, Zhang et al., 2023a, inter alia) to report micro F1 with NoTA relation excluded. Following previous work (Zhang et al., 2023a, Li et al., 2023c) and to keep OpenAI API costs under control, we randomly select 1,000 examples from each dataset’s test partition to serve as our test set.

Baselines For small language model-based methods, we selected two low-resource state-of-the-art methods: $\text{NLI}_{\text{DeBERTa}}$ (Sainz et al., 2021) and $\text{SuRE}_{\text{PEGASUS}}$ (Lu et al., 2022). For LLMs baselines, we selected QA4RE (Zhang et al., 2023a) and SUMASK (Li et al., 2023c). We also evaluated the performance of a Vanilla prompting method. Further details of the baselines are as follows.

- $\text{NLI}_{\text{DeBERTa}}$ (Sainz et al., 2021) reformulates RE as a natural language inference (NLI) task and uses a DeBERTa model that is finetuned on MNLI dataset as the entailment engine.

Method	TACRED			TACREV			Re-TACRED			SemEval			Avg
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
NLI$_{\text{DeBERTa}}$†	42.9	76.9	55.1	43.3	84.6	57.2	71.7	58.3	64.3	22.0	25.7	23.7	50.1
SuRE$_{\text{PEGASUS}}$†	13.8	51.7	21.8	13.5	54.1	21.6	16.6	34.6	22.4	0.0	0.0	0.0	16.4
Vanilla‡	35.7	51.6	37.9	42.5	77.8	55.0	62.0	81.5	70.5	57.1	63.2	60.0	55.9
QA4RE‡	38.0	82.1	52.0	38.7	85.8	53.4	65.2	91.9	76.3	40.1	48.2	43.8	56.4
SUMASK‡	42.5	76.8	54.7	43.9	80.2	56.8	62.5	73.4	67.5	16.0	15.7	15.8	48.7
Ours‡	56.0	72.5	**63.2**	58.9	82.6	**68.8**	75.5	79.8	**77.6**	80.6	83.8	**82.1**	**72.9**

Table 1: Main results on four RE datasets. We mark the best results in bold. † marks re-implemented results from Zhang et al. (2023a). ‡ denotes our runs with GPT4.1.

LLM	TACRED	TACREV	RETACRED	SemEval	Avg
Gemma 3 27B	57.4	63.6	67.4	65.2	63.4
Llama 3.1 70B	57.3	63.9	73.1	73.8	67.0
Mistral Large 2411	63.2	69.1	72.4	73.2	69.5
GPT4o mini	56.1	66.2	69.0	64.0	63.8
GPT4.1	63.2	68.8	77.6	82.1	72.9

Table 2: Evaluation of our method using various open source and proprietary LLMs on the four RE datasets.

- SuRE$_{\text{PEGASUS}}$ (Lu et al., 2022) reformulates RE as a summarization task and utilizes PEGASUS$_{\text{Large}}$ obtaining competitive results in few-shot and fully-supervised settings.

- QA4RE (Zhang et al., 2023a) reformulates RE as multiple-choice question answering in order to take advantage of QA's higher prevalence in instruction-tuning training data of LLMs.

- SUMASK (Li et al., 2023c) for each relation type, generates a set of summarizations and yes/no questions, and then asks a LLM to answer the yes/no questions based on the summarizations. Then performs a computation on the answers using an uncertainty based active learning method to estimate output probabilities of the LLM.

- Vanilla Prompt (Zhang et al., 2023a) is a simple and direct prompt strategy. We use the version from QA4RE authors.

We ran all LLM baselines as well as our method with the same LLM: GPT4.1. We also evaluated our method with various LLMs, namely Gemma 3 27B, Llama 3.1 70B, Mistral Large 2411 (123B), and GPT4o mini. The details of the implementation of our method are provided in Appendix E.

4.2 Results

Our evaluation of zero-shot relation extraction on the four RE datasets is shown in Table 1. Our Relation-Aware Prompting technique outperform SOTA methods in all four datasets. Our method provides significant improvements of 8.5 F1 points on TACRED, 12 points on TACREV, 1.3 points on RETACRED, and 22.1 points on SemEval. The improvements on SemEval are important because the dataset has been known to be more challenging for zero-shot methods due to (1) lack of entity typing, (2) relations being undirected, and (3) overlapping relations between the same entity mentions. These results are highly encouraging considering that our method relies solely on off-the-shelf LLMs and no additional components. We also evaluated our method with various open source and proprietary LLMs shown in Table 2. While bigger models perform slightly better, our method works across all LLMs. Even our method evaluated on Gemma 27B outperforms prompting baselines such as QA4RE ans SUMASK that are evaluated with GPT4.1 on three out of four datasets, even though Gemma is orders of magnitude smaller than GPT4.1.

4.3 Ablation Study

We conduct an ablation study to analyze the effectiveness of the proposed elements of our method. The experiments were run on a subset of the development partitions of TACRED and SemEval. We randomly sampled 1000 examples from the development sets. We selected GPT4o mini and Gemma 3 27B to conduct the experiments. In each experiment we remove an element of our main prompt and report the results. In each experiment the number of synthesized examples is a hyperparameter chosen from {0, 1, 5, 10} via hyperparam-

Prompts	TACRED		SemEval	
	4o mini	Gemma3	4o mini	Gemma3
Main Prompt	**61.9**	**65.5**	**62.6**	**66.1**
w/o Rel. Defs.	60.9	62.6	56.4	58.9
w/o Targeted Sum.	57.2	58.1	–	–
w/o Reduc. Label Space	45.1	46.5	58.5	63.1

Table 3: Ablation study on TACRED and SemEval.

eter search.

Table 3 shows the results. We observe that removing "Relation Definitions and Annotation Guidelines", "Targeted Summarization", and "Reducing the Label Space" from our final prompt decreases the performance considerably (as mentioned before, we do not do targeted summarization for SemEval), reaffirming the effectiveness of the proposed components.

5 Conclusion

In this work, we present Relation-Aware Prompting, an effective prompt-based method for zero-shot relation extraction. We propose targeted summarization of instances with respect to the underlying relations to bring out the relations in the texts, reducing the applicable relations through a method inspired by the process of elimination, synthesizing examples using subject-verb-object structure, and other prompting elements. We evaluate our approach on four RE datasets. Our approach significantly outperforms current zero-shot LLM prompt-based methods. Our approach can also be considered as a new strong baseline for zero-shot RE that is compatible with any LLM.

Limitations

We conduct comprehensive experiments exclusively on zero-shot RE and showed that our approach is a new, robust state-of-the-art method. However, we did not engage in few-shot RE, domain-specific explorations, or other languages. Thus, the performance of our method on these settings is still unclear. We acknowledge these matters and leave answering these questions for future work.

Acknowledgments

The authors would like to thank Guozheng Li and other authors of SUMASK prompting work for providing us the source code of their original work.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Alberto Cetoli. 2020. Exploring the zero-shot limit of FewRel. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1447–1451, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online. Association for Computational Linguistics.

Amir DN Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. Relation classification as two-way span-prediction. *arXiv preprint arXiv:2010.04829*.

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

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Jiaying Gong and Hoda Eldardiry. 2021. Prompt-based zero-shot relation classification with semantic knowledge augmentation. *arXiv preprint arXiv:2112.04539*.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022a. Generative prompt tuning for relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3170–3185, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2024. An empirical study on information extraction using large language models. *Preprint*, arXiv:2305.14450.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022b. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Huang. 2023. Evaluation of ChatGPT on biomedical tasks: A zero-shot comparison with fine-tuned generative transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 326–336, Toronto, Canada. Association for Computational Linguistics.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *Preprint*, arXiv:2304.11633.

Bo Li, Dingyao Yu, Wei Ye, Jinglei Zhang, and Shikun Zhang. 2023b. Sequence generation with label augmentation for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13043–13050.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023c. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6877–6892, Singapore. Association for Computational Linguistics.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024. Meta in-context learning makes large language models better zero and few-shot relation extractors. *Preprint*, arXiv:2404.17807.

Junpeng Li, Zixia Jia, and Zilong Zheng. 2023d. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. 2022. Summarization as indirect supervision for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6575–6594, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389, Singapore. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Mahdi Rahimi and Mihai Surdeanu. 2023. Improving zero-shot relation classification via automatically-acquired entailment templates. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 187–195, Toronto, Canada. Association for Computational Linguistics.

Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. *Transactions of the Association for Computational Linguistics*, 9:691–706.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. *Preprint*, arXiv:2310.03668.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13843–13850.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, and Guoyin Wang. 2023. Pushing the limits of chatgpt on nlp tasks. *Preprint*, arXiv:2306.09719.

Anushka Swarup, Tianyu Pan, Ronald Wilson, Avanti Bhandarkar, and Damon Woodard. 2025. LLM4RE: A data-centric feasibility study for relation extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6670–6691, Abu Dhabi, UAE. Association for Computational Linguistics.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *Preprint*, arXiv:2303.04360.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547, Singapore. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021a. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021b. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023a. S2ynRE: Two-stage self-training with synthetic data for low-resource relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8186–8207, Toronto, Canada. Association for Computational Linguistics.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023b. How to unleash the power of large language models for few-shot relation extraction? In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 190–200, Toronto, Canada (Hybrid). Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *Preprint*, arXiv:2303.10420.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023a. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada. Association for Computational Linguistics.

Wenjie Zhang, Xiaoning Song, Zhenhua Feng, Tianyang Xu, and Xiaojun Wu. 2023b. Labelprompt: Effective prompt-based learning for relation classification. *arXiv preprint arXiv:2302.08068*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Sizhe Zhou, Yu Meng, Bowen Jin, and Jiawei Han. 2024. Grasping the essentials: Tailoring large language models for zero-shot relation extraction. *Preprint*, arXiv:2402.11142.

A Related Work

A.1 Pre-LLM Works

Prior to the advent of large language models, most recent approaches for supervised relation extraction use pretrained masked language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) or adapt sequence-to-sequence models to the task, such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). Traditional RE methods needed a large amount of labeled data for training models from scratch (Kambhatla, 2004, Zeng et al., 2014). The pre-LLM recent approaches outperform traditional approaches by finetuning a masked language model (Wu and He, 2019, Joshi et al., 2020, Yamada et al., 2020, Wang et al., 2021b, Lyu and Chen, 2021, Paolini et al., 2021, Wang et al., 2022, Li et al., 2023b) or prompting a masked language model (Han et al., 2022b, Han et al., 2022a, Zhang et al., 2023b).

As for low-resource RE, several approaches have been proposed for relation extraction with few training examples (Han et al., 2018, Gao et al., 2019, Baldini Soares et al., 2019, Sabo et al., 2021). For the problem of zero-shot RE, approaches leverage techniques such as similarity based Siamese architectures (Chen and Li, 2021) and indirect supervision as task reformulation. In the literature, zero-shot RE has been reformulated as other tasks such as reading comprehension (Levy et al., 2017), textual entailment (Sainz et al., 2021, Rahimi and Surdeanu, 2023), summarization (Lu et al., 2022), span-prediction (Cohen et al., 2020), question answering (Cetoli, 2020), triple generation (Wang et al., 2022, Wang et al., 2021a), and prompting (Gong and Eldardiry, 2021).

A.2 LLM-based Works

Since our work is a LLM-based approach, we focus the rest of the section on similar LLM-based methods for RE with special focus on zero-shot methods. A common approach is prompting LLMs for data generation and then use the generated data to finetune a small language model (Xu et al., 2023b, Zhou et al., 2024, Xu et al., 2023a, Tang et al., 2023). Another common approach is finetuning a retriever to retrieve relevant training examples to be used as in-context learning demonstrations (Sun et al., 2023, Wan et al., 2023). We do not use finetuning or retrieval in our approach. The rest of the common approaches is as follows.

LLM Prompt-Based Methods: In addition to vanilla prompting (Ye et al., 2023, Li et al., 2023a, Ma et al., 2023, Jahan et al., 2023), several approaches have been proposed. Li et al. (2023c) is a zero-shot method that for each relation type, generates a set of summarizations and yes/no questions, and then asks a LLM to answer the yes/no questions based on the summarizations. Then performs a complex computation on the answers using an uncertainty based active learning method to estimate output probabilities of the LLM. Wei et al. (2024) turns zero-shot IE tasks including entity-relation triple extraction into an interactive dialogue-like multiple turns QA. Zhang et al. (2023a) reformulates RE as multiple-choice question answering in order to take advantage of QA's higher prevalence in instruction-tuning training data of LLMs. In this method, manually-constructed relation verbalization templates are used to generate the options of multiple-choice questions. Agrawal et al. (2022) uses a guided prompt design to direct the LLM towards a structured output for clinical relation extraction. Our approach is different from these approaches as we do not require complex post-prompt computations or interactive dialogue-like QA or guided prompt design.

Methods That Use Annotation Guidelines: Zhou et al. (2024) uses annotation guidelines to prompt a LLM to generate synthetic data and then trains a small language model with this data for zero-shot RE. Sainz et al. (2024) uses annotation guidelines to finetune a large language model for IE tasks. They puts annotation guidelines, input and gold output in the prompt to finetune the LLM. Then use the LLM to perform zero-shot IE on unseen datasets. Pang et al. (2023) does not use guidelines, but rather learns them and then use them for prompting LLMs. They automatically synthesize a set of guidelines based on a few error cases, and during inference retrieve helpful guidelines for better classification. Li et al. (2023d) integrates a LLM and a natural language inference (NLI) module to generate relation triples. They use relation descriptions to construct hypotheses for NLI and to guide NLI to output expected relations. Our approach is different from these approaches as we only use guidelines for our prompt without finetuning or using NLI.

Methods That Finetune LLMs: Wadhwa et al. (2023) finetunes a T5 model using Chain of Thought style explanations generated by GPT-3.

Li et al. (2024) uses a meta-training framework for zero and few-shot RE by tuning a LLM to perform in-context learning on 12 RE datasets, and then evaluate it on unseen RE benchmarks. Wang et al. (2023) proposes a unified information extraction framework, and reformulates IE tasks to the sequence-to-sequence form and solves them through fine-tuning LLMs. Our approach is different from these methods as we don't finetune LLMs.

Summarization: Li et al. (2023c) produces k targeted summarizations, questions, and answers for each relation type. Then the vector representations of these items are generated and used to estimate the conditional probabilities for each relation type. Instead, we use targeted summarization once and place it directly in our final prompt. Lu et al. (2022) reformulates RE as a summarization task. They convert input sentences with an entity information verbalization technique and convert output relations with label verbalization templates. Then with the converted inputs and outputs that suit a summarization model, they adopt such a model. The model is pretrained on summarization tasks and then simply finetuned with the converted inputs and outputs. This method is different from ours as: (a) it requires finetuning of a summarization model whereas ours is zero-shot; and (b) the summary output are the verbalization templates whereas ours are more natural.

B Dataset Statistics

The statistics of the datasets are shown in Table 4.

Dataset	# train	# dev	# test	# rel.
TACRED	68,124	22,631	15,509	42
TACREV	68,124	22,631	15,509	42
RETACRED	58,465	19,584	13,418	40
SemEval	8,000	-	2,717	19

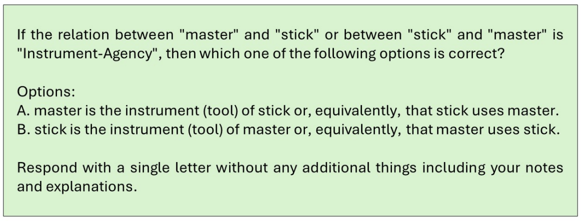
Table 4: Statistics of TACRED, TACREV, RETACRED and SemEval.

C Targeted Summarization Example

Complex sentences can confuse LLMs due to misleading cues. Figure 3 (top) shows an example. For this example, ChatGPT predicts the relation of "other family" (a family relation other than immediate family), but the gold label is "no relation". The presence of some cues in the sentence such as the word "family" may have confused the model. The prediction of the model on the summarized version, however, is correct.

D Determining the Direction of a Relation

If relations are undirected, as in SemEval 2010 Task 8 dataset, one extra step is required to determine the direction of the relation. To this end, we prompt the LLM to choose the directionality of the relation from two options that are created from a template. The template is chosen from the very first sentence of the relation definitions. For instance, for "Instrument-Agency" we use the following sentence as the template: `X is the instrument (tool) of Y or, equivalently, that Y uses X`. We create two sentences with the template. For one sentence, we replace "X" with the head entity and replace "Y" with the tail entity. For the other sentence, we swap the entities. Finally, we use the two sentences as options of a multiple-choice question in the prompt. Figure 2 shows the prompt.



If the relation between "master" and "stick" or between "stick" and "master" is "Instrument-Agency", then which one of the following options is correct?

Options:
A. master is the instrument (tool) of stick or, equivalently, that stick uses master.
B. stick is the instrument (tool) of master or, equivalently, that master uses stick.

Respond with a single letter without any additional things including your notes and explanations.

Figure 2: Our prompt for selecting the direction of a relation in SemEval dataset.

E Implementation Details of Our Method

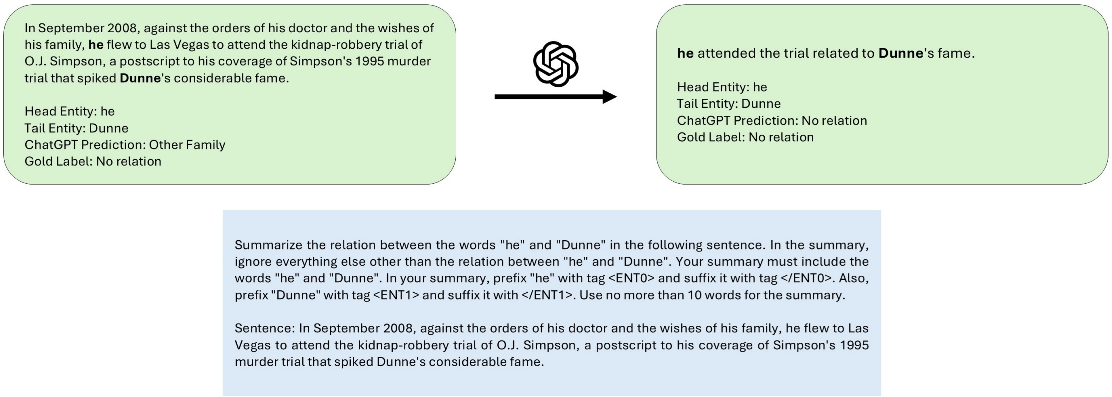
TACRED, TACREV, and RETACRED datasets provide entity types. Therefore, we use entity type constraints to reduce applicable label space. SemEval dataset, however, is focused on common nouns. For SemEval we use our proposed prompt-based method to reduce applicable label space.

SemEval sentences are short. Therefore, we do not use Targeted Summarization for SemEval. For TACRED, TACREV, and RETACRED we use it. However, there are some examples in these datasets where the head and tail entities have identical text in a case-insensitive way (e.g. "He" and "he" in the sentence "He told the Times he no longer is active in the Church of Scientology"). For these instances, we skip the summarization as we thought it could confuse the models.

In our experiments, we set temperature to zero. Our hyperparameters are every element of our prompt, such as the number of synthesized examples, whether to use summarization, whether to use entity tagging, etc. These hyperparameters are selected using a small set equal to 1% of development set. This set contains a few examples per relation. This setting is comparable to using examples in the annotation guidelines as development.

F Prompts

In this section, we present our prompts for Targeted Summarization (Figure 3), Reducing Label Space (Figure 4), Synthesizing Examples (Figure 5), and our final prompt (Figure 6).



In September 2008, against the orders of his doctor and the wishes of his family, **he** flew to Las Vegas to attend the kidnap-robbery trial of O.J. Simpson, a postscript to his coverage of Simpson's 1995 murder trial that spiked **Dunne**'s considerable fame.

Head Entity: he
Tail Entity: Dunne
ChatGPT Prediction: Other Family
Gold Label: No relation

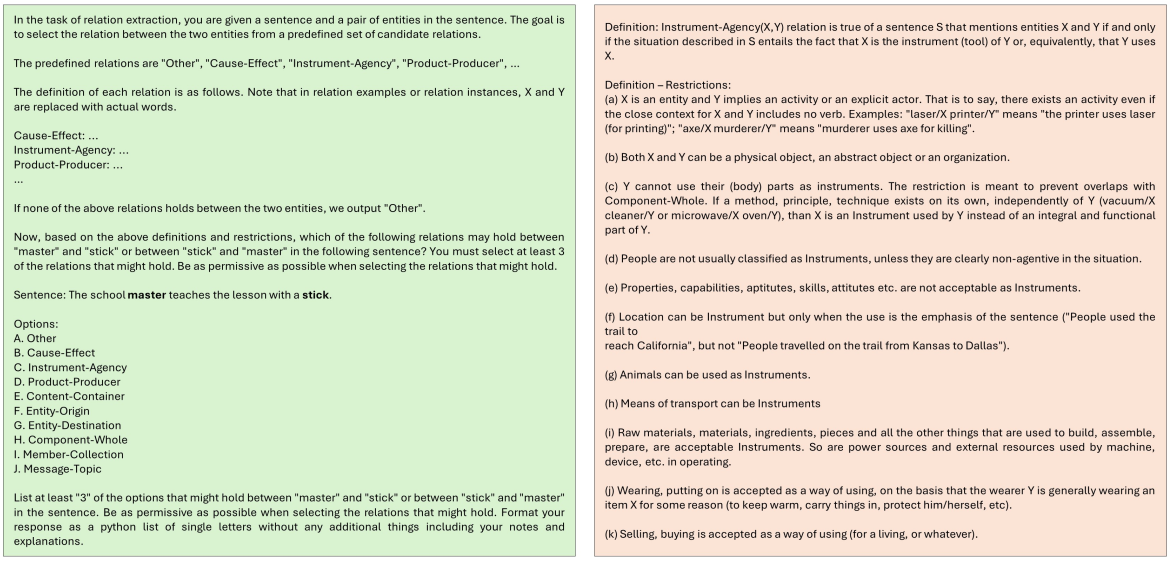
he attended the trial related to **Dunne**'s fame.

Head Entity: he
Tail Entity: Dunne
ChatGPT Prediction: No relation
Gold Label: No relation

Summarize the relation between the words "he" and "Dunne" in the following sentence. In the summary, ignore everything else other than the relation between "he" and "Dunne". Your summary must include the words "he" and "Dunne". In your summary, prefix "he" with tag <ENT0> and suffix it with tag </ENT0>. Also, prefix "Dunne" with tag <ENT1> and suffix it with </ENT1>. Use no more than 10 words for the summary.

Sentence: In September 2008, against the orders of his doctor and the wishes of his family, he flew to Las Vegas to attend the kidnap-robbery trial of O.J. Simpson, a postscript to his coverage of Simpson's 1995 murder trial that spiked Dunne's considerable fame.

Figure 3: Top: Targeted summarization helps relation extraction. ChatGPT predicts the incorrect relation "other family" (a family relation other than immediate family) when the original text is used, but the gold label is "no relation" (top left). The presence of some cues in the sentence such as the word "family" may have confused the model. The prediction of the model on the summarized version, however, is correct (top right). Bottom: Our prompt for summarizing the text supporting the relation between the entities.



In the task of relation extraction, you are given a sentence and a pair of entities in the sentence. The goal is to select the relation between the two entities from a predefined set of candidate relations.

The predefined relations are "Other", "Cause-Effect", "Instrument-Agency", "Product-Producer", ...

The definition of each relation is as follows. Note that in relation examples or relation instances, X and Y are replaced with actual words.

Cause-Effect: ...
Instrument-Agency: ...
Product-Producer: ...
...

If none of the above relations holds between the two entities, we output "Other".

Now, based on the above definitions and restrictions, which of the following relations may hold between "master" and "stick" or between "stick" and "master" in the following sentence? You must select at least 3 of the relations that might hold. Be as permissive as possible when selecting the relations that might hold.

Sentence: The school **master** teaches the lesson with a **stick**.

Options:
A. Other
B. Cause-Effect
C. Instrument-Agency
D. Product-Producer
E. Content-Container
F. Entity-Origin
G. Entity-Destination
H. Component-Whole
I. Member-Collection
J. Message-Topic

List at least "3" of the options that might hold between "master" and "stick" or between "stick" and "master" in the sentence. Be as permissive as possible when selecting the relations that might hold. Format your response as a python list of single letters without any additional things including your notes and explanations.

Definition: Instrument-Agency(X,Y) relation is true of a sentence S that mentions entities X and Y if and only if the situation described in S entails the fact that X is the instrument (tool) of Y or, equivalently, that Y uses X.

Definition – Restrictions:
(a) X is an entity and Y implies an activity or an explicit actor. That is to say, there exists an activity even if the close context for X and Y includes no verb. Examples: "laser/X printer/Y" means "the printer uses laser (for printing)"; "axe/X murderer/Y" means "murderer uses axe for killing".

(b) Both X and Y can be a physical object, an abstract object or an organization.

(c) Y cannot use their (body) parts as instruments. The restriction is meant to prevent overlaps with Component-Whole. If a method, principle, technique exists on its own, independently of Y (vacuum/X cleaner/Y or microwave/X oven/Y), than X is an Instrument used by Y instead of an integral and functional part of Y.

(d) People are not usually classified as Instruments, unless they are clearly non-agentive in the situation.

(e) Properties, capabilities, aptitutes, skills, attitutes etc. are not acceptable as Instruments.

(f) Location can be Instrument but only when the use is the emphasis of the sentence ("People used the trail to
reach California", but not "People travelled on the trail from Kansas to Dallas").

(g) Animals can be used as Instruments.

(h) Means of transport can be Instruments

(i) Raw materials, materials, ingredients, pieces and all the other things that are used to build, assemble, prepare, are acceptable Instruments. So are power sources and external resources used by machine, device, etc. in operating.

(j) Wearing, putting on is accepted as a way of using, on the basis that the wearer Y is generally wearing an item X for some reason (to keep warm, carry things in, protect him/herself, etc).

(k) Selling, buying is accepted as a way of using (for a living, or whatever).

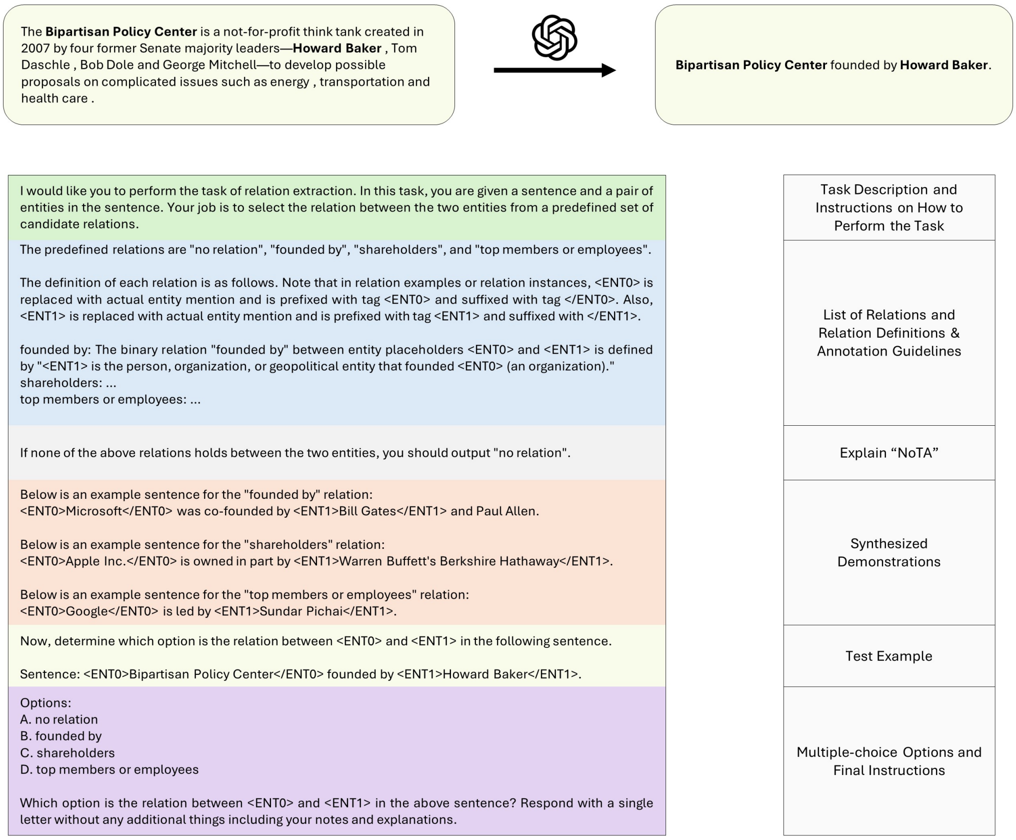
Figure 4: Left: Our prompt that reduces the number of candidate relation types for SemEval from 10 to 3. Parts of the prompt omitted for brevity. Right: The part of annotation guidelines that we selected to use for Instrument-Agency relation in the prompt.

Please give 10 examples of the relation "founded by" between a organization (called head entity) and a person (called tail entity) using a single subject-verb-object structure containing the head entity and the tail entity.

Note that the tail entity is the person, organization, or geopolitical entity that founded the assigned organization.

Prefix the head entity with tag <ENT0> and suffix it with tag </ENT0>. Also, prefix the tail entity with tag <ENT1> and suffix it with tag </ENT1>.
Produce your response as a list of strings in a json list object.

Figure 5: Our prompt for synthesizing example sentences for relations. In this example, the relation is “founded by”.



The **Bipartisan Policy Center** is a not-for-profit think tank created in 2007 by four former Senate majority leaders—**Howard Baker** , Tom Daschle , Bob Dole and George Mitchell—to develop possible proposals on complicated issues such as energy , transportation and health care .

Bipartisan Policy Center founded by **Howard Baker**.

I would like you to perform the task of relation extraction. In this task, you are given a sentence and a pair of entities in the sentence. Your job is to select the relation between the two entities from a predefined set of candidate relations.

The predefined relations are "no relation", "founded by", "shareholders", and "top members or employees".

The definition of each relation is as follows. Note that in relation examples or relation instances, <ENT0> is replaced with actual entity mention and is prefixed with tag <ENT0> and suffixed with tag </ENT0>. Also, <ENT1> is replaced with actual entity mention and is prefixed with tag <ENT1> and suffixed with </ENT1>.

founded by: The binary relation "founded by" between entity placeholders <ENT0> and <ENT1> is defined by "<ENT1> is the person, organization, or geopolitical entity that founded <ENT0> (an organization)."
shareholders: ...
top members or employees: ...

If none of the above relations holds between the two entities, you should output "no relation".

Below is an example sentence for the "founded by" relation:
<ENT0>Microsoft</ENT0> was co-founded by <ENT1>Bill Gates</ENT1> and Paul Allen.

Below is an example sentence for the "shareholders" relation:
<ENT0>Apple Inc.</ENT0> is owned in part by <ENT1>Warren Buffett's Berkshire Hathaway</ENT1>.

Below is an example sentence for the "top members or employees" relation:
<ENT0>Google</ENT0> is led by <ENT1>Sundar Pichai</ENT1>.

Now, determine which option is the relation between <ENT0> and <ENT1> in the following sentence.

Sentence: <ENT0>Bipartisan Policy Center</ENT0> founded by <ENT1>Howard Baker</ENT1>.

Options:
A. no relation
B. founded by
C. shareholders
D. top members or employees

Which option is the relation between <ENT0> and <ENT1> in the above sentence? Respond with a single letter without any additional things including your notes and explanations.

Task Description and Instructions on How to Perform the Task
List of Relations and Relation Definitions & Annotation Guidelines
Explain “NoTA”
Synthesized Demonstrations
Test Example
Multiple-choice Options and Final Instructions

Figure 6: Top: The relation between the head and tail entities in a test example is summarized by an LLM. Bottom: The structure of our final prompt. The prompt uses the summarized example. The example is chosen from the TACRED dataset.

Enhancing Readability-Controlled Text Modification with Readability Assessment and Target Span Prediction

Fengkai Liu, John S. Y. Lee
Department of Linguistics and Translation
City University of Hong Kong
Hong Kong SAR, China
`fengkaliu3-c@my.cityu.edu.hk, jsylee@cityu.edu.hk`

Abstract

Readability-controlled text modification aims to rewrite an input text so that it reaches a target level of difficulty. This task is closely related to automatic readability assessment (ARA) since, depending on the difficulty level of the input text, it may need to be simplified or complexified. Most previous research in LLM-based text modification has focused on zero-shot prompting, without further input from ARA or guidance on text spans that most likely require revision. This paper shows that ARA models for texts and sentences, as well as predictions of text spans that should be edited, can enhance performance in readability-controlled text modification.

1 Introduction

Readability-controlled text modification aims to rewrite the input text so that it reaches a target level of difficulty (Chi et al., 2023; Farajidizaji et al., 2024). The nature of the modification depends on whether the difficulty level of the input text is higher, lower, or the same as the target level. If higher, the system should perform *text simplification*, which has been extensively studied (Agrawal and Carpuat, 2023; Alva-Manchego et al., 2020b; Mo and Hu, 2024; Štajner et al., 2017). If lower, the system needs to "increase the lexical and syntactic complexity of a text" (Berov and Standvoss, 2018), a task that has been studied under various terms including "textual embellishment" (Berov and Standvoss, 2018), "text elaboration" (Farajidizaji et al., 2024) and "complexification" (Chi et al., 2023). In Table 1, for example, the input text is simplified to the target level "elementary" and complexified to the "advanced" level, but is left unchanged when the target level is "intermediate", since it is already at the intermediate level.

Input (intermediate level)	Argentina is unhappy that the US outdoor clothing retailer, Patagonia, is claiming a domain name that has been known far longer as a region of spectacular beauty that also has its own parliament ...
Output for target level "elementary"	Argentina is unhappy that the US outdoor clothing retailer, Patagonia *wants* a domain name that ...
Output for target level "intermediate"	Argentina is unhappy that the US outdoor clothing retailer, Patagonia, is claiming a domain name that has been known far longer as a region of spectacular beauty that ...
Output for target level "advanced"	Argentina has *lodged an expression of its unhappiness* that the US outdoor clothing retailer, Patagonia, is claiming a domain name that ...

Table 1: Example input and outputs for readability-controlled text modification at the target levels "elementary", "intermediate", and "advanced"

In the most recent study on readability-controlled text modification (Farajidizaji et al., 2024), zero-shot prompting of Large Language Models (LLMs) has been shown capable of nudging the difficulty level of a text towards the target level, in terms of the Flesch Reading-Ease Score (FRES) (Kincaid et al., 1975). There was no attempt, however, to further improve the proposed method using training data from text modification or automatic readability assessment (ARA) (Martinc et al., 2021), which is directly relevant to readability-controlled text modification. Since an ARA model estimates the difficulty level of a text, it can help determine whether and how much modification is needed, and in which parts of the input text.

This paper investigates whether ARA and text

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 293–303
November 8-9, 2025 ©2025 Association for Computational Linguistics

modification training data can improve the performance of readability-controlled text modification. Specifically, we address the following research questions:

ARA data Can text modification performance be improved with automatic assessment of the difficulty of a text and/or of individual sentences?

Text modification data Can text modification performance be improved with automatic identification of text spans that require editing?

The rest of the paper is organized as follows. After a review of previous research (Section 2), we describe the proposed auxiliary models for guiding the text modification process (Section 3), and the prompt that incorporates information from these auxiliary models (Section 4). We then present our dataset (Section 5) and report automatic (Section 6) and human (Section 7) evaluation results.

2 Previous work

Readability-controlled text modification may be viewed as a paraphrasing task (Barzilay and Lee, 2003) with an additional constraint, namely, that the output must be at the target level of difficulty. To our knowledge, there have been only three published studies on text modification. Trott and Rivière (2024) assessed the ability of GPT-4 Turbo to make a text easier or harder, but did not require the output to reach a specific difficulty level. Chi et al. (2023) focused on text modification at the sentence level. T5 models were fine-tuned on sentence pairs from text simplification datasets; for sentence complexification, T5 models were fine-tuned with the inputs and outputs reversed. Most related to our work, Farajidizaji et al. (2024) generated versions of the input text at different target FRES, using zero-shot prompting of ChatGPT and Llama-2. In terms of the readability of the modified texts, their best model achieved 24.2% accuracy in reaching the target FRES. In terms of the content of the modified texts, however, no evaluation was reported because of the lack of gold outputs.

Other studies have focused exclusively on either text complexification or text simplification. For the former task, Berov and Standvoss (2018) trained an LSTM model on the inverse of WikiLarge, a simplification corpus based on simple and standard English Wikipedia. The model is then applied to complexify the texts in a story corpus. Naskar et al. (2019) adopted a similar LSTM encoder-decoder, and reported both BLEU and human evaluation.

For text simplification, early studies tended to take the relative approach, i.e., to make the input text easier but without specifying a target difficulty level or audience (Belder and Moens, 2010; Kajiwara et al., 2013; Paetzold and Specia, 2017; Siddharthan, 2002). Recent work has increasingly recognized the need for text simplification to an absolute target level (Štajner et al., 2017). Nishihara et al. (2019) used lexical and syntactic complexity features, while Yanamoto et al. (2022) applied deep reinforcement learning using a reward calculated based on the difference between the difficulty of the output sentence and the target difficulty. Agrawal et al. (2021) used a non-autoregressive model to iteratively edit the source sentence. Agrawal and Carpuat (2023) predicts low-level control tokens for text simplification. Similar to this work, more recent studies have exploited LLMs. For example, the *SimplifyMyText* system rewrites the input text in plain language (Färber et al., 2025). In the *ExpertEase* system, LLM-based agents collaborate in text simplification playing the roles of the expert, the teacher, and the student (Mo and Hu, 2024).

3 Approach

Readability-controlled text modification requires judgment on text difficulty, and on the kinds of content that are most suitable for revision. Therefore, it may potentially benefit from auxiliary models that can assess the difficulty of sentences (Section 3.1) and texts (Section 3.2), and predict the text spans that require revision (Section 3.3). For LLM-based text modification, the information produced by these auxiliary models can be incorporated into the prompt (Section 4).

3.1 Sentence ARA (Sent ARA) Model

Generally, it is not necessary to rewrite every sentence in a text, even when transforming the text to a distant target difficulty level. To make judicious changes, it could be useful to highlight the sentences that most likely require revision, i.e., the easiest or the most complex sentences.

A sentence-level ARA model predicts the complexity of an individual sentence (Brunato et al., 2018; Garbacea et al., 2021; Liu et al., 2025; Lu et al., 2020; Schicchi et al., 2020; Štajner et al., 2017). This information can help guide the LLM in identifying sentences that deviate most from the

With the owners out of the cats' line of vision, researchers played recordings of three strangers calling the cats' names followed by a call from the cat's owner and then by the call of another stranger. `advanced`
Researchers charted the cats' reactions by measuring a number of responses, including head movements, tail and ear movements, eye dilation and vocalization or whether they moved their paws. `advanced`
When strangers called their names, the cats had no reaction to the voices whatsoever. `intermediate`

Table 2: Example output of the Sent ARA Model (Section 3.1), which labels each sentence in the input text with its difficulty level.

*With *the *owners *out *of *the *cats *' *line *of *vision *, *researchers *played *recordings *of *three *strangers *calling *the *cats *' *names *followed *by *a *call *from *the *cat *'s *owner *and *then *by *the *call *of *another *stranger *. Researchers charted the cats ' reactions by measuring a number of responses *, *including head movements , tail and ear movements , eye dilation and *vocalization *or whether they moved their paws . When strangers called their names , the cats had no reaction to the voices whatsoever .

Table 3: Example output of the Target Span Prediction Model (Section 3.3): words predicted by the model to require revision are marked with asterisks.

target difficulty level. Table 2 shows an example assessment, which labels the level of each sentence in a text as "elementary", "intermediate", or "advanced".

3.2 Text ARA Model

As opposed to sentence-level ARA, a text-level ARA model (Martinc et al., 2021) assesses the overall difficulty level of a text. This model can identify input texts that are already at the target difficulty level, and therefore do not require any modification. Further, since the LLM may not be able to modify the input text to its target level in one round (Farajidizaji et al., 2024), this model can also determine the necessity of an additional round of modification.

3.3 Target Span Prediction Model

As a preliminary step before text modification, it could be useful to identify the words that require revision (Chen and Meurers, 2019; Collins-Thompson, 2014; Liu et al., 2024). For example, complex word identification can serve as the first step in a text simplification pipeline to identify the *target words*, i.e., the difficult vocabulary items that should be replaced (Gooding and Kochmar, 2019; Paetzold and Specia, 2016; Shardlow, 2014).

In the context of text modification, we will use the term *target span* to refer to the parts of the input text that should be edited. Example input-output pairs of text simplification and complexification can be used for training a tagger that predicts these spans. Compared to the Sent ARA Model (Section 3.1), this model can provide more fine-grained guidance. In the example shown in Table 3, both sentences and individual words (e.g., "including") have been predicted to require editing.

4 Prompt implementation

The system prompt (Table 9 in Appendix A.1) describes the text modification task. The average FRES (Kincaid et al., 1975) is provided for each difficulty level, since the LLM may not be familiar with the difficulty scale.

Table 4 shows the user prompt and the auxiliary models (Section 3) from which the content is derived. The prompt states both the target difficulty level, as well as the difficulty level of the source text ("elementary", "intermediate", or "advanced"), as estimated by the Text ARA Model (Section 3.2).

In the input text, asterisks are placed on each word that is predicted to require editing, according to the Target Span Prediction Model (Section 3.3). Finally, each sentence in the input text is shown with its difficulty level, as estimated by the Sent ARA Model (Section 3.1).

As demonstrations, two sample input texts are shown with their gold output text and gold predictions from the Sent ARA, Text ARA and Target Span Prediction Models[1].

4.1 Proposed prompts

The following variations of the proposed prompt were implemented:

SentARA+Span The full prompt (Table 4).

---

[1] Using the actual rather than gold predictions from these models led to slightly worse performance.

Prompt template	Model
Rewrite the following `<text_ara_output>` passage at the `<target_level>` level. Tokens starting with a '*' symbol in the source text indicate the words that were changed from the source passage to the rewritten passage. `<demonstrations>`	Text ARA
Source passage: `<target_span_output>`	Target Span Prediction
Difficulty of individual sentences in source passage: `<sent_ara_output>` Rewritten passage:	Sent ARA

Table 4: Prompt for text modification (left), which makes use of information provided by the auxiliary models (right): `<target_level>` is the target difficulty level; `<text_ara_output>` is the predicted level of the input text; `<target_span_output>` is the input text with target span predictions (Table 3); `<sent_ara_output>` is the predicted level of each sentence in the input text (Table 2); `<demonstrations>` are the two sample input/output pairs

SentARA Information from the Target Span Prediction Model is omitted.

SentARA+Span+Ling The full prompt, with linguistic features (Section 4.3) added.

SentARA+Span$^{\times n}$ The full prompt, iteratively issued until the Text ARA Model predicts the input text has reached the target level, up to a maximum of n iterations.

4.2 Chain-of-thought prompts

Chain-of-thought (CoT) guides LLMs in generating their own intermediate steps for completing a task. CoT reasoning has led to robust performance in multiple NLP tasks (Brown et al., 2020; Chen et al., 2019; Ling et al., 2017; Wei et al., 2022). We implemented the following CoT prompts:

CoT The prompt uses the instruction "Let's think step by step" (Kojima et al., 2022) to obtain the reasoning for text modification. The LLM-generated reasoning is then included after the input text in the prompt, which does not use any information from the auxiliary models.

CoT (zero-shot) Same as above, except that no sample input/outputs are provided.

4.3 Baseline prompts

Linguistic features have been shown to be effective for text simplification (Agrawal et al., 2021; Maddela et al., 2021; Mo and Hu, 2024; Nishihara et al., 2019; Yanamoto et al., 2022). To identify the most salient features, we extracted all available features in the Lexical Complexity Analyzer (Lu, 2012) and calculate their correlation with the difficulty levels of the text in our training data. According to the Pearson Correlation Coefficient (Table 10 in Appendix A.2), the top 5 features are Root Text-to-Token Ratio (TTR), which measures lexical diversity; Corrected TTR, a refined version of TTR accounting for text length; Number of Different Words, which counts the number of unique words; the Uber Index, a composite, holistic measure for lexical complexity and diversity; and L2, the proportion of content words. The following baseline prompts were implemented:

Ling This prompt includes only the statement: "The measurements of the five linguistic features of the source passages are `<ling_feats>`", where `<ling_feats>` refers to the five features mentioned above.

Vanilla No linguistic feature or auxiliary model is used.

Vanilla (Zero-shot) Same as above, and no sample input/outputs are given.

5 Data

Newsela (Xu et al., 2015) is a graded parallel corpus derived from 1,911 news articles. For each article, simplified versions have been composed by professional editors for students between Grade 2 and Grade 12.

To facilitate learning of the revision patterns across the spectrum of difficulty levels, for each of these 1,911 articles, we retrieved three versions that span the grades: one version between Grades 2 and 5, which we will refer to as the "elementary" level (`ele`); one version between Grades 6 and 8, the "intermediate" level (`int`); and one version between Grades 9 and 12, the "advanced" level (`adv`).

In our experiments, each of these 5,733 texts is to be revised to all three target levels {`ele`,`int`,`adv`}. Since the level of the source text is not disclosed to the system, it does not know whether the text should be simplified (`int->ele`, `adv->ele`, `adv->int`), complexified (`ele->int`, `ele->adv`, `int->adv`), or left unchanged (`ele->ele`, `int->int`, `adv->adv`).

6 Automatic Evaluation

6.1 Implementation details

We used `Meta-Llama-3.1-8B-Instruct`[2] for all prompts (Section 4). The auxiliary models were implemented as follows:

Sent ARA Model Following the approach proposed by Liu and Lee (2023), we trained a BART-large (Lewis et al., 2019) model to classify a sentence [3] at the `ele`, `int`, or `adv` level.

Text ARA Model Following the approach proposed by Lee et al. (2021), we trained a neural ARA model by fine-tuning BART (Lewis et al., 2020) on the Newsela dataset.[4]

Target Span Prediction Model We trained a tagger to label each word in the source text as `REVISE` or `KEEP`. As shown in Table 3, the words tagged as `REVISE` will be asterisked in the source text in the prompt. The gold labels were derived from the sentence-aligned text pairs from Newsela (Section 5), with sentence alignments automatically produced by SentAlign [5] (Steingrímsson et al., 2023). All words in a source sentence that are not in the aligned target sentence are considered `REVISE`; all words in a source sentence that is not aligned to any target sentence are also considered `REVISE`, since they are deleted. We trained six separate RoBERTa-based (Liu et al., 2019) sequence taggers[6] to cover all combinations of source and target levels ({`ele`,`int`,`adv`}).

We also attempted training these three models using several other transformers, but did not produce any significant improvement in performance.

6.2 Evaluation metrics

The output text should have the target complexity and appropriate content. Complexity is evaluated with two metrics. The first, Mean Absolute Error (**MAE**) in FRES, is the difference between the FRES of the output text and the FRES of the gold text. We also report **Accuracy**, i.e. whether the output text is at the target level of difficulty. The difficulty level of the output text is estimated with the Text ARA Model (Section 3.2).

The content quality of the output text is evaluated with four metrics. To determine the degree of meaning preservation, the semantic similarity between the source text and output text is evaluated using **BERTScore** (Zhang et al., 2019). Further, the output text is compared against the gold text using three widely adopted metrics in text simplification evaluation: **BLEU** (Papineni et al., 2002)[7], **SARI** (Xu et al., 2016)[8], and **D-SARI** (Sun et al., 2021) which aims at document-level simplification[9].

6.3 Results

6.3.1 Auxiliary models

To construct the prompt for each input text (Table 4), we obtained outputs from the auxiliary models (Section 3) using 5-fold cross validation. The Text ARA Model achieved an accuracy of 98.87% on the three-way classification of difficulty level (`ele`, `int`, or `adv`). The high accuracy validates its reliability as an evaluation metric for text complexity (Section 6.2). In contrast, the use of FRES, even with score thresholds optimized on the Newsela dataset, would yield only 66.08% accuracy on the three-way classification of difficulty level.

The Sent ARA Model performed at 0.680 accuracy and 0.674 F1-score on the three-way classification of difficulty level for sentences. The Target Span Prediction Model attained 0.454 precision and

---

[2] `meta-llama/Meta-Llama-3.1-8B-Instruct`

[3] We used the `BartForSequenceClassification model` from the transformers library of HuggingFace (Wolf et al., 2020)

[4] We used the pre-trained base version of BART from Huggingface (Wolf et al., 2020).

[5] `https://github.com/steinst/SentAlign`

[6] We used the `RobertaForTokenClassification model` from HuggingFace (Wolf et al., 2020).

[7] The NLTK (Bird, 2006) implementation was used.

[8] The EASSE simplification evaluation suite (Alva-Manchego et al., 2019) was used.

[9] The implementation by Sun et al. (2021) (`https://github.com/RLSNLP/Document-level-text-simplification`) was used. include D-SARI incorporates several penalty factors, in addition to the add, keep and delete scores in SARI. The same weights for these scores were used for both text simplification and development.

Aux. Model	Prompt method	D-SARI	SARI	BLEU	BertScore	Accuracy	MAE ↓
Nil	Vanilla (zero-shot)	16.18	43.35	39.73	0.878	0.512	16.83
	CoT (zero-shot)	17.90	43.38	45.18	0.875	0.588	16.36
	Vanilla	15.08	43.22	41.74	0.890	0.584	11.80
	CoT	18.32	42.96	50.70	0.890	0.614	10.76
	Ling	16.05	42.51	47.08	0.902	0.487	8.52
Sent ARA	SentARA	17.48	41.20	45.33	0.901	**0.720**	9.19
Sent ARA and Target Span Prediction	SentARA+Span	21.38	**49.39**	52.28	0.904	0.609	**7.74**
	SentARA+Span+Ling	20.34	46.48	51.30	**0.908**	0.559	7.83
	SentARA+Span$^{\times 2}$	**22.69**	49.22	**52.80**	0.902	0.688	7.74
	SentARA+Span$^{\times 3}$	22.66	49.16	52.72	0.902	0.698	7.80

Table 5: Text modification performance using different auxiliary models (↓ means smaller is better)

Task	Prompt	P	R
Overall	Vanilla (zero-shot)	0.640	0.479
	SentARA	0.642	0.549
	SentARA+Span	**0.694**	**0.603**
Simp.	Vanilla (zero-shot)	0.644	0.482
	SentARA	0.639	0.547
	SentARA+Span	**0.665**	**0.568**
Comp.	Vanilla (zero-shot)	0.635	0.476
	SentARA	0.645	0.551
	SentARA+Span	**0.719**	**0.634**

Table 6: Performance in identifying text spans to edit (without regard to the quality of the final output), with breakdown into simplification (simp.) and complexification (comp.)

0.650 recall when simplifying texts, and 0.448 precision and 0.657 recall when complexifying texts.

6.3.2 Effect of Sentence ARA

Table 5 presents experimental results on text modification. When none of the auxiliary models is used ("Nil" row in Table 5), CoT prompting gave the best Accuracy (0.614), D-SARI (18.32) and BLEU (50.70) scores, though it was outperformed by the CoT (zero-shot) prompt in terms of SARI. Both of these CoT prompts improved performance over their vanilla version. Consistent with Wei et al. (2022) and Kojima et al. (2022), the self-generated reasoning steps were helpful in guiding the LLM in performing text modification. The linguistic features (Section 4.3) led to the best result in terms of MAE (8.52) and BertScore (0.902).

The use of the Sent ARA Model led to the highest Accuracy (0.720). To better understand the effect of this auxiliary model, we measured its precision and recall in identifying text spans for revision, without considering the quality of the actual revision. As shown in Table 6, the gains of the Sent ARA prompt over the Vanilla baseline were mostly due to the recall (0.549 vs. 0.479). This suggests that the ARA predictions helped the LLM in selecting sentences for revision that were missed by the baseline.

However, the use of the Sent ARA Model did not generally improve the quality of the modified text. It was outperformed by the CoT and Ling prompts in most metrics other than Accuracy.

6.3.3 Effect of Target Span Prediction

Incorporating predictions of the target spans (SentARA+Span) resulted in the best overall performance (Table 5). These predictions helped produce output texts that resembled the gold texts to a greater extent. In terms of the content, SentARA+Span attained higher D-SARI, SARI and BLEU scores compared to all baselines. As shown in Table 6, it yielded higher precision (0.694) and recall (0.603) than SentARA, likely because it was able to make judicious choices in selecting individual words for revision, whereas SentARA provided guidance only at the sentence level. In terms of complexity, it also produced outputs that are closest to the gold texts in FRES (7.74 MAE). However, it was outperformed by the SentARA prompt on Accuracy. This suggests that, if the overriding objective is to achieve the target difficulty level, then the use of sentence ARA alone could be worth considering.

Although the addition of linguistic features (SentARA+Span+Ling) further increased BERTScore, it did not help improve the quality of text modification on the other metrics. This suggests that the LLM may have difficulty interpreting the linguistic features and their implications.

6.3.4 Effect of Text ARA

Initial modification. As discussed in Section 6.3.1,

Prompt method	Task	D-SARI	SARI	BLEU	BertScore	Accuracy	MAE ↓
SentARA+Span	Simp.	**22.84**	46.30	45.21	0.892	0.440	8.42
	Comp.	19.91	**52.47**	**59.34**	**0.916**	**0.777**	**7.06**
SentARA+Span (intermediate only)	Simp.	**25.42**	55.81	**59.74**	**0.928**	**0.685**	7.14
	Comp.	20.07	**60.97**	58.32	0.913	0.675	**6.14**

Table 7: Breakdown of text modification performance into simplification (Simp.) and complexification (Comp.), based on (top) all input texts; (bottom) intermediate input texts only (↓ means smaller is better)

the Text ARA Model has a much higher accuracy in estimating the difficulty level of a text than the use of FRES. For text modification, it was thus effective in determining whether the input text is already at the target difficulty level, or requires modification. Only 1.13% of the input texts underwent unnecessary modification; conversely, only 0.50% of the input texts failed to undergo modification, due to incorrect estimation from this model. The corresponding percentages would have been 33.92% and 24.95%, respectively, if FRES were used for this purpose.

Iterative modifications. In terms of the content of the modified texts, an additional round of modification (SentARA+Span$^{\times 2}$) further improved the BLEU and D-SARI scores. A third iteration (SentARA+Span$^{\times 3}$), however, led to a slight decline in performance, likely because repeated modifications may exacerbate the biases inherent in LLMs (Gallegos et al., 2024; Yu et al., 2024). In terms of text complexity, while the Accuracy improved as expected, a second iteration had no effect on MAE (7.74) and a third one led to negative impact (7.80).

6.3.5 Simplification vs. complexification

To analyze the differences between text simplification and complexification, we compare the performance of the best prompt (SentARA+Span) on these two tasks. As shown in the top of Table 7, complexification appears to be an easier task than simplification, offering better performance on all but one metric (D-SARI). Complexification often requires inserting new content at appropriate places in a text, which could be more challenging and subjective than removing existing content in simplification. This may explain its lower score for D-SARI, which puts more emphasis on the quality of document-level organization.

A potential confounding factor is the length of the input text. The input texts that required simplification were on average longer, since they were originally at the intermediate and advanced levels; those that required complexification were shorter, since they were taken from the intermediate and elementary levels. To avoid this bias, the bottom of Table 7 considers only the input texts at the intermediate level. Simplification now offers better performance in terms of BLEU and BERTScore. This may be due to the wider array of choices when selecting more complex words or sentence structures, in comparison to selecting simpler ones. In terms of complexity, the gap between the two tasks is narrower for both Accuracy and MAE, but more research is needed to explain the difference.

7 Human evaluation

7.1 Evaluation metrics

The quality of text modification was evaluated by two human judges, a master and a PhD student in Linguistics. Similar to previous schemes (Alva-Manchego et al., 2020a; Yang et al., 2023), Fluency and Meaning were annotated on a 5-point Likert scale (1=Strongly disagree, 5=Strongly agree):

- **Fluency**: The output text is fluent and free of grammatical errors.
- **Meaning**: The output text adequately preserves the meaning of the source text.

To accommodate both simplification and complexification, Complexity was scored from -5 to +5:

- **Complexity**: A score of +5 means the output text is much more complex and harder to understand than the source input; 0 means they are comparable in complexity; and -5 means the output text is much easier.

7.2 Evaluation set-up

Two source texts were randomly selected at each difficulty level (`ele`, `int` and `adv`). Each of these six texts was paired with four modified versions, namely, its modified version at the two other levels as produced by the Vanilla (zero-shot) and

Modification type	Source->target level	Fluency		Meaning		Complexity	
		Zero-shot	Proposed	Zero-shot	Proposed	Zero-shot	Proposed
Text Simp.	Overall	4.00	**4.17**	3.17	**3.50**	-3.50	-3.50
	`adv->int`	4.00	4.00	3.50	**4.00**	**-3.00**	-2.50
	`adv->ele`	4.00	4.00	3.00	3.00	-3.50	**-4.00**
	`int->ele`	4.00	**4.50**	3.00	**3.50**	-4.00	-4.00
Text Comp.	Overall	**3.50**	3.17	4.00	**4.17**	+3.00	**+3.50**
	`int->adv`	**3.50**	3.00	4.50	**5.00**	+3.50	**+4.00**
	`ele->int`	**4.00**	3.00	3.00	**3.50**	**+3.00**	+2.50
	`ele->adv`	3.00	**3.50**	**4.50**	4.00	+2.50	**+4.00**

Table 8: Human evaluation scores on Fluency, Meaning and Complexity (Section 7)

SentARA+Span$^{\times 2}$ models. The two human judges independently scored these 24 text pairs on Complexity, Fluency and Meaning (Section 7.1).

7.3 Results

The human evaluation results are shown in Table 8. The two judges achieved a Cohen's kappa (Cohen, 1968) of 0.67 for Fluency, 0.88 for Meaning, and 0.85 for Complexity, all at/above a substantial level of agreement.

Meaning. The proposed model (SentARA+Span$^{\times 2}$) achieved a higher Meaning score than the zero-shot baseline in four of the six settings. It was slightly outperformed in `adv->ele` and tied at `ele->adv`. As both of these settings required greater modification (distance of two levels rather than one), the proposed model was more likely to make changes that altered the original meaning.

Fluency. The proposed model performed better in simplification, but it slightly underperformed in complexification, particularly with two-level modifications. While the auxiliary models help the proposed model in modifying the content and difficulty, they do not necessarily improve the fluency of the output text, which sometimes contains awkward phrasing.

Complexity. Both models were capable of revising the input text towards the required complexity level, obtaining positive Complexity scores when the level of the input text was lower than the target (complexification needed), and negative scores when its level was higher than the target (simplification needed). When simplifying `adv` texts, the proposed model succeeded in differentiating between the target levels `ele` and `int`, reducing text complexity to a much greater degree for the former (-4.00) than the latter (-2.50). A smaller difference was observed for the zero-shot baseline (-3.50 vs. -3.00). When complexifying `ele` texts, the proposed model was again able to make a sharper distinction, producing a more sophisticated output for the `adv` target grade (+4.00) than for `int` (+2.50). The zero-shot model failed to do so and produced an `adv` output (+2.50) that is easier than the `int` output (+3.00).

8 Conclusion

Human editors often need to tailor a text for readers at different proficiency levels. Readability-controlled text modification aims to rewrite an input text so that it reaches a target level of difficulty. Depending on the difficulty of the input text, it may need to be simplified or complexified. This paper has presented the first study on LLM-based readability-controlled text modification that leverages ARA and prediction of target spans, i.e. the parts of the input text that require editing.

We trained ARA models that can predict the difficulty level of a sentence or a text, and taggers that predict whether each word should be revised. The information from these auxiliary models are then incorporated into the prompt for the LLM. Experimental results on the Newsela corpus showed that both the ARA models and the target span prediction model improved the quality of the modified text. In future work, we plan to evaluate this approach on other text genres, and investigate whether fine-tuning an LLM on text modification data can lead to further performance gains.

Acknowledgments

We gratefully acknowledge support from the eLearning Ancillary Facilities Programme of the Quality Education Fund (Project "Knowledge Overlord - A Self-sustaining AI Game-based Online Platform to Enhance Student's Literacy Ability and

21st Century Skills"); and from the Language Fund of the Standing Committee on Language Education and Research (project EDB(LE)/P&R/EL/203/14).

References

Sweta Agrawal and Marine Carpuat. 2023. Controlling pre-trained language models for grade-specific text simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 12807–12819.

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. Easse: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*.

J. De Belder and M. F. Moens. 2010. Text Simplification for Children. In *Proc. SIGIR Workshop on Accessible Search Systems*.

Leonid Berov and Kai Standvoss. 2018. Discourse embellishment using a deep encoder-decoder network. In *Proc. 3rd Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2018)*.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

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

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699.

Xiaobin Chen and Detmar Meurers. 2019. Linking text readability and learner proficiency using linguistic complexity feature vector distance. *Computer Assisted Language Learning*, 32(4):418–447.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V Le. 2019. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Alison Chi, Li-Kuang Chen, Yi-Chen Chang, Shu-Hui Lee, and Jason S. Chang. 2023. Learning to Paraphrase Sentences to Different Complexity Levels. *Transactions of the Association for Computational Linguistics*, 11:1332–1354.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339.

M. Färber, P. Aghdam, K. Im, M. Tawfelis, and H. Ghoshal. 2025. SimplifyMyText: An LLM-Based System for Inclusive Plain Language Text Simplification. *Advances in Information Retrieval. ECIR 2025. Lecture Notes in Computer Science*, 15575.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1086–1097.

Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting Proper Lexical Paraphrase for Children. In *Proc. 25th Conference on Computational Linguistics and Speech Processing (ROCLING)*, pages 59–73.

Peter J. Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas for Navy enlisted personnel. In *Research Branch Report 8–75*. Chief of Naval Technical Training: Naval Air Station Memphis.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Bruce W. Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, page 7871–7880.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Fengkai Liu, Yishi Jiang, Chun Lai, and Tan Jin. 2024. Teacher engagement with automated text simplification for differentiated instruction.

Fengkai Liu, Tan Jin, and John S. Y. Lee. 2025. Automatic readability assessment for sentences: neural, hybrid and large language models. *Language Resources and Evaluation*, 59:2265–2296.

Fengkai Liu and John S. Y. Lee. 2023. Hybrid Models for Sentence Readability Assessment. In *Proc. 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, page 448–454.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dawei Lu, Xinying Qiu, and Yi Cai. 2020. Sentence-level readability assessment for l2 chinese learning. In *Chinese Lexical Semantics: 20th Workshop, CLSW 2019, Beijing, China, June 28–30, 2019, Revised Selected Papers 20*, pages 381–392. Springer.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553.

Matej Martinc, Senja Pollak, Marko, and Robnik-Šikonja. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179.

Kaijie Mo and Renfen Hu. 2024. Expertease: A multi-agent framework for grade-specific document simplification with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9080–9099.

Subhajit Naskar, Soumya Saha, and Sreeparna Mukherjee. 2019. Text embellishment using attention based encoder-decoder model. In *Proc. 4th Workshop on Computational Creativity in Language Generation*, page 28–38.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, pages 260–266.

Gustavo H Paetzold and Lucia Specia. 2016. Plumberr: An automatic error identification framework for lexical simplification. In *Proceedings of the first international workshop on Quality Assessment for Text Simplification (QATS)*, pages 1–9.

Gustavo H. Paetzold and Lucia Specia. 2017. Lexical Simplification with Neural Ranking. In *Proc. EACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Daniele Schicchi, Giovanni Pilato, and Giosué Lo Bosco. 2020. Deep neural attention-based model for the evaluation of italian sentences complexity. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 253–256. IEEE.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *LREC*, pages 1583–1590.

Advaith Siddharthan. 2002. An Architecture for a Text Simplification System. In *Proc. Language Engineering Conference (LEC).*

Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic assessment of absolute sentence complexity. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI*, volume 17, pages 4096–4102.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2023. SentAlign: Accurate and Scalable Sentence Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Singapore, Singapore. Association for Computational Linguistics.

Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. Document-level text simplification: Dataset, criteria and baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013.

Sean Trott and Pamela Rivière. 2024. Measuring and Modifying the Readability of English Texts with GPT-4. In *Proc. 3rd Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, page 126–134.

Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic Assessment of Absolute Sentence Complexity. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI).*

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, page 38–45.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwara, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. Controllable text simplification with deep reinforcement learning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 398–404.

Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023. A new dataset and empirical study for sentence simplification in Chinese. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8306–8321, Toronto, Canada. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

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

A Appendix

A.1 System prompt

You are a professional expert in modifying texts into a target difficulty level. The Flesch Reading-Ease Score (FRES) measures the readability of the text:

The averaged FRES for elementary texts is `<aver_ele_FRES>`.
The averaged FRES for intermediate texts is `<aver_int_FRES>`.
The averaged FRES for advanced texts is `<aver_adv_FRES>`.

Table 9: System prompt

A.2 Linguistic Features

Rank	Feature	Correl.
1	Root TTR	0.684
2	Corrected TTR	0.684
3	Number of Different Words	0.628
4	Uber Index	0.610
5	LS2	0.546

Table 10: The five lexical features that are most correlated with readability levels of the texts in our dataset.

TAG–EQA: Text–And–Graph for Event Question Answering via Structured Prompting Strategies

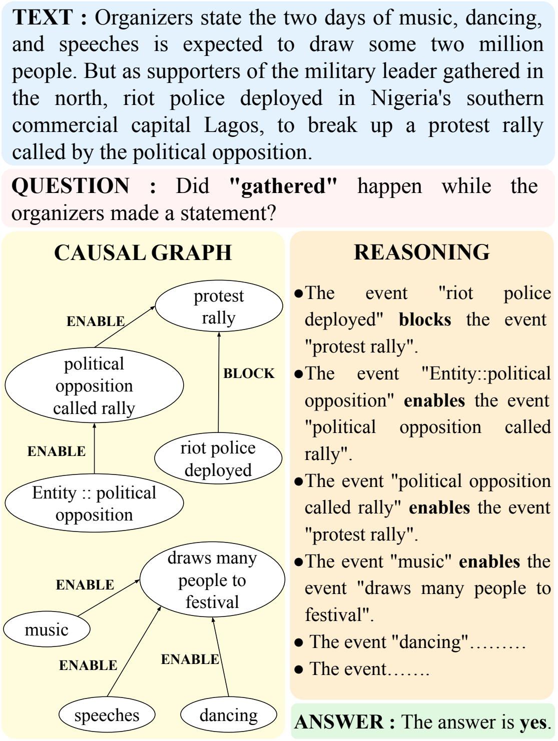
Maithili Kadam **Francis Ferraro**
University of Maryland, Baltimore County
`mkadam1@umbc.edu`, `ferraro@umbc.edu`

Abstract

Large language models (LLMs) excel at general language tasks but often struggle with event-based questions—especially those requiring causal or temporal reasoning. We introduce **TAG-EQA** (Text-And-Graph for Event Question Answering), a prompting framework that injects causal event graphs into LLM inputs by converting structured relations into natural-language statements. TAG-EQA spans nine prompting configurations, combining three strategies (zero-shot, few-shot, chain-of-thought) with three input modalities (text-only, graph-only, text+graph), enabling a systematic analysis of when and how structured knowledge aids inference. On the TORQUESTRA benchmark, TAG-EQA improves accuracy by 5% on average over text-only baselines, with gains up to 12% in zero-shot settings and 18% when graph-augmented CoT prompting is effective. While performance varies by model and configuration, our findings show that causal graphs can enhance event reasoning in LLMs without fine-tuning, offering a flexible way to encode structure in prompt-based QA.[1]

1 Introduction

Consider the text in Figure 1: *"Organizers state the two days of music, dancing, and speeches is expected to draw two million people. But as supporters gathered... riot police deployed..."*. When asked, *"Did the protesters* GATHER *while the organizers* MADE A STATEMENT*?"* , answering correctly requires chaining events: $music \rightarrow draw\_crowd \rightarrow gather$, while recognizing that $riot\_police\_deployed \dashv organizers\_state$, where $\rightarrow$ denotes an "enables" relation and $\dashv$ denotes a "blocks" relation.



TEXT : Organizers state the two days of music, dancing, and speeches is expected to draw some two million people. But as supporters of the military leader gathered in the north, riot police deployed in Nigeria's southern commercial capital Lagos, to break up a protest rally called by the political opposition.

QUESTION : Did **"gathered"** happen while the organizers made a statement?

REASONING

- The event "riot police deployed" **blocks** the event "protest rally".
- The event "Entity::political opposition" **enables** the event "political opposition called rally".
- The event "political opposition called rally" **enables** the event "protest rally".
- The event "music" **enables** the event "draws many people to festival".
- The event "dancing"………
- The event…….

ANSWER : The answer is **yes**.

Figure 1: Illustrative example from the TORQUESTRA dataset. **Top:** Narrative passage and a binary event–based question. **Left:** Annotated causal graph showing ENABLE and BLOCK relations between events. **Right:** A step–by–step reasoning trace that follows the graph to support causal inference. Together, the graph and reasoning highlight how structured event relations enable models to answer questions that require indirect causal chaining.

Such questions require structured event reasoning, where causal graphs make dependencies explicit by surfacing ENABLE and BLOCK relations that go beyond surface cues (Regan et al., 2023; Chambers and Jurafsky, 2008; Dunietz et al., 2020; Jain et al., 2023; Chi et al., 2024). Without structure, LLMs often rely on shallow lexical patterns and miss deeper event logic.

We explore how structured causal knowledge can guide large language models in reasoning about events. Specifically, we introduce **TAG–EQA**–**T**ext–**A**nd–**G**raph for **E**vent **Q**uestion **A**nswering–a prompting framework that converts

---

[1] Code and data available at `https://github.com/MaithiliKadam4/TAG-EQA`

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 304–315
November 8-9, 2025 ©2025 Association for Computational Linguistics

causal event graphs into natural language cues and embeds them directly into the prompt. Rather than fine–tuning the model, TAG–EQA steers its inference by aligning causal structure with prompt format, enabling models to reason more coherently about event dynamics. It spans nine prompting configurations, combining three strategies (zero–shot, few–shot, and chain–of–thought) with three input modalities (*text–only*, *graph–only*, and *text+graph*). While this space is broad, our analysis reveals that causal graphs are especially effective when paired with reasoning-oriented prompts such as chain–of–thought. See Section 3 for full details.

In our experiments on the TORQUESTRA dataset (Regan et al., 2023), TAG–EQA improves accuracy by approximately 5% over text-only baselines, with gains rising to 12% in zero-shot and 18% in chain-of-thought settings. To better understand where structure helps, we group questions into thirteen semantic categories—such as *causal*, *temporal*, and *hypothetical* reasoning—and find that graph-based prompts are particularly effective for *causal chains*, *temporal dependencies*, and *counterfactual what-if scenarios*, where structured event interactions are central to answering correctly. Because these experiments rely on gold human-annotated graphs, the reported numbers should be interpreted as an upper bound on the benefit of structured input; robustness to automatically induced or noisy graphs remains future work.

Our contributions are as follows:

- We introduce *TAG-EQA*, a prompting framework that incorporates causal event graphs into LLM inputs via natural-language serialization—without requiring model fine-tuning.
- We evaluate nine prompting configurations across three strategies and three input types, using T5-XXL, Qwen-32B, and GPT-3.5/4o.[2]
- We examine how causal graphs and reasoning traces interact, and when they improve model performance.
- We report accuracy trends across thirteen semantic question types to identify where structured and/or reasoning-based input helps the most.

2 Related Work

Prior work on event modeling, causal reasoning, and prompt engineering has independently advanced narrative QA. We synthesize these strands by embedding structured causal graphs into prompt formats to guide event-centric inference in LLMs.

2.1 Event Modeling

Narrative understanding has long relied on modeling event relations such as causality, enablement, and sequence. Early work induced event chains using verb–argument frames (Chambers and Jurafsky, 2008), while later approaches inferred causal links from raw text without explicit structure (Dunietz et al., 2020). TORQUESTRA (Regan et al., 2023) builds on this by aligning QA pairs with human-annotated causal graphs, enabling evaluation of structured reasoning in context.

We build on these efforts by treating enable and block relations as first-class prompt components. Each edge is serialized into a natural language sentence, allowing LLMs to ground their reasoning in structured temporal and causal dependencies.

2.2 Cause-Effect Graphs and Causal Reasoning

Causal reasoning from text remains a significant challenge for large language models (LLMs), which often conflate correlation with causation (Yamin et al., 2024). Early methods extracted causal links using pattern-based heuristics (Radinsky et al., 2012), while later approaches employed pretrained language models to infer implicit dependencies from raw text (Dunietz et al., 2020). More recent work has shown that explicitly incorporating cause–effect graphs can improve question answering on narrative and commonsense tasks (Roy et al., 2024; Bethany et al., 2024). However, most prior efforts emphasize direct or temporal links, leaving finer-grained structures underutilized.

However, enabling (A enables B) and blocking (C blocks D) relations remain underexplored despite their value in modeling conditional constraints and counterfactuals. We address this by formalizing them into natural-language prompts that explicitly guide LLM reasoning.

2.3 Prompt Engineering and Chain-of-Thought Reasoning

Prompt engineering enables pretrained language models to perform new tasks without parameter updates, leveraging Zero- and Few- shot in-context learning (Petroni et al., 2019; Brown et al., 2020). Chain-of-thought (CoT) prompting extends this approach by encouraging step-by-step reasoning

---

[2] GPT-3.5 is used for non-reasoning prompts (Zero and Few), while GPT-4o is used for reasoning (CoT) due to its stronger multi-step inference ability.

through natural language traces (Wei et al., 2022). Enhancements such as self-consistency decoding and automatic CoT generation, aim to improve reliability and reduce dependence on handcrafted examples (Wang et al., 2023; Zhang et al., 2023). Although CoT prompting has shown strong results in arithmetic and symbolic tasks (Wei et al., 2022; Kojima et al., 2022), its use in structured, event-based inference remains limited. We explore this intersection by aligning CoT prompts with causal graphs—letting models reason over explicitly structured event dynamics across prompt formats.

3 Method

TAG-EQA investigates whether structured causal knowledge and explicit reasoning can improve event-based question answering (QA) when delivered through prompt design. We vary two orthogonal factors: (1) the *prompting strategy*—Zero, Few, or CoT, and (2) the *input modality*—Text, Graphs, or TAG (text and graph combined). This results in nine prompting configurations, each combining a reasoning style with one or more input sources. We evaluate these configurations across three instruction-tuned LLMs (T5-XXL, Qwen-32B, and GPT-3.5/4o) to understand how prompt structure and content influence QA accuracy.
Figure 2 provides a visual overview of our prompting pipeline. Prompts are constructed by combining a narrative passage, a natural-language representation of a causal graph (if present), and optionally, demonstration QA examples or intermediate reasoning traces. See Section 3.3 for full details.

3.1 Task Formulation

Each instance consists of a short passage P, a yes/no question Q about events in P, and optionally a causal event graph G–either an *instance* or *schema* graph–encoding directed ENABLES/BLOCKS dependencies. An ENABLES edge ($A \rightarrow B$) indicates that event A provides a prerequisite or supportive condition for event B to occur, while a BLOCKS edge ($C \dashv D$) denotes that event C prevents, interrupts, or otherwise inhibits event D. The model must output "yes" or "no." In CoT prompts, it must first produce a natural-language reasoning trace, then the final answer.

3.2 Dataset: TORQUESTRA

We use the TORQUESTRA dataset (Regan et al., 2023) to construct prompts for event-based QA grounded in causal and temporal structure. Each instance provides a short narrative passage, a yes/no question, and one or more directed causal graphs with `ENABLES`/`BLOCKS` edges. We generate prompts for all nine configurations by combining QA pairs with the corresponding passage and/or a verbalized version of the graph (i.e., each edge serialized into a natural-language sentence such as "Event A enables Event B"), formatted according to the selected prompting strategy (Zero, Few, or CoT) and input modality (Text, Graphs, or TAG).

Track Name	Strategy	Modality	Avg. Prompt Length	Reason Length
Zero–Text	Zero	Text	95.2	–
Zero–Graphs	Zero	Graphs	80.6	–
Zero–TAG	Zero	TAG	138.0	–
Few–Text	Few	Text	121.7	–
Few–Graphs	Few	Graphs	178.0	–
Few–TAG	Few	TAG	242.8	–
CoT–Text	CoT	Text	229.2	30.7
CoT–Graphs	CoT	Graphs	287.6	30.7
CoT–TAG	CoT	TAG	336.8	30.7

Table 1: **Prompt lengths for each TAG–EQA track:** Prompt lengths (tokens) across the three strategies (Zero, Few, CoT) and input modalities (Text, Graphs, TAG). CoT prompts include explicit reasoning traces.

All prompts are derived from the human-refined subset ($\text{TORQUESTRA}_{\text{human}}$), which provides gold-standard causal graphs. Figure 1 illustrates a typical example: the passage, graph, and question are used to build the prompt, although the figure content is for exposition only and not used verbatim.

Our filtered Full split contains 477,549 QA instances, balanced across strategies and input types. To support ablations and cost-sensitive models, we also define a Small subset of 1,024 instances, stratified by question category and prompting configuration. Unless otherwise noted, results are reported on the Full set, with Small results shown separately for GPT-based models.

Prompt length varies considerably by configuration. For example, Zero–Text prompts average 95.2 tokens, while CoT–TAG prompts reach 336.8 tokens on average, with reasoning traces contributing 30.7 tokens. These differences affect both model performance and context length constraints.

As shown in Table 1, 26.5% of answers are "yes" and 73.5% are "no." Each causal graph omits approximately 5.3 events on average, requiring inference over missing links—a key motivation for evaluating the utility of structured prompts.

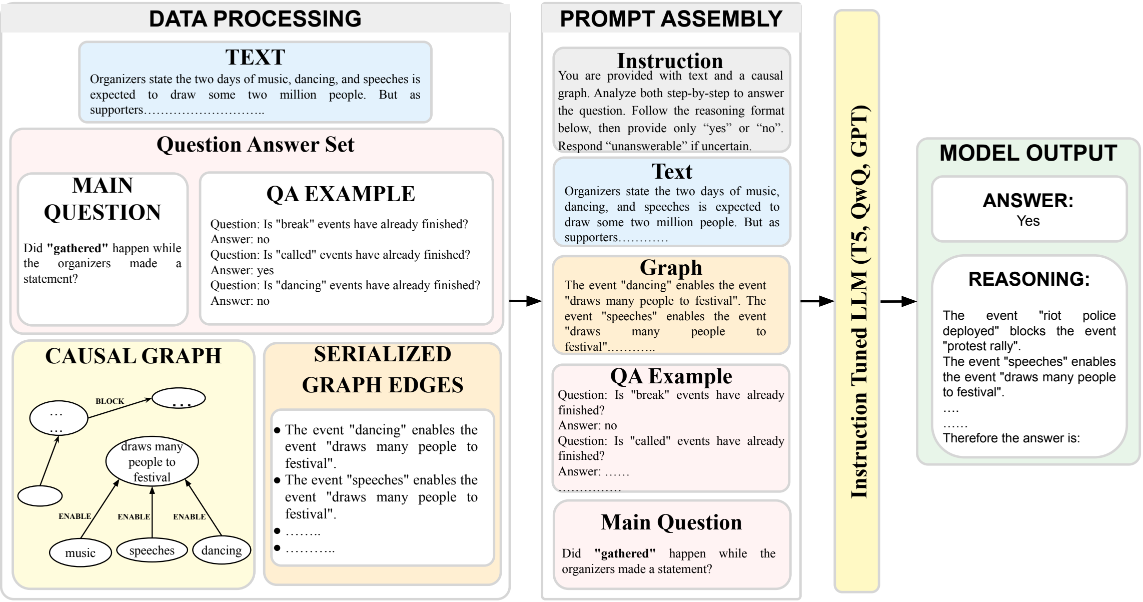

DATA PROCESSING
TEXT
Organizers state the two days of music, dancing, and speeches is expected to draw some two million people. But as supporters……………………..
Question Answer Set
MAIN QUESTION
Did "gathered" happen while the organizers made a statement?
QA EXAMPLE
Question: Is "break" events have already finished?
Answer: no
Question: Is "called" events have already finished?
Answer: yes
Question: Is "dancing" events have already finished?
Answer: no
CAUSAL GRAPH
BLOCK
draws many people to festival
ENABLE
ENABLE
ENABLE
music
speeches
dancing
SERIALIZED GRAPH EDGES
The event "dancing" enables the event "draws many people to festival".
The event "speeches" enables the event "draws many people to festival".
PROMPT ASSEMBLY
Instruction
You are provided with text and a causal graph. Analyze both step-by-step to answer the question. Follow the reasoning format below, then provide only "yes" or "no". Respond "unanswerable" if uncertain.
Text
Organizers state the two days of music, dancing, and speeches is expected to draw some two million people. But as supporters…………
Graph
The event "dancing" enables the event "draws many people to festival". The event "speeches" enables the event "draws many people to festival"………..
QA Example
Question: Is "break" events have already finished?
Answer: no
Question: Is "called" events have already finished?
Answer: ……
Main Question
Did "gathered" happen while the organizers made a statement?
Instruction Tuned LLM (T5, QwQ, GPT)
MODEL OUTPUT
ANSWER:
Yes
REASONING:
The event "riot police deployed" blocks the event "protest rally".
The event "speeches" enables the event "draws many people to festival".
Therefore the answer is:

Figure 2: **Overview of our QA prompting pipeline for TAG + CoT configuration.** From left to right: a narrative passage and associated causal graph are processed into a structured input. The causal graph is serialized into natural-language edges (yellow), and the original passage text is retained (blue). Prompt assembly combines task instructions, the text, the graph, in-context QA examples, and the main question into a single input to the instruction-tuned LLM (T5, QwQ, or GPT). The model produces both a yes/no answer and a step-by-step reasoning trace grounded in the causal structure (green).

3.3 Prompt-Track Configurations

TAG-EQA combines three prompting strategies with three input modalities, yielding a 3×3 grid of nine prompt configurations (e.g., Zero–Text, Few–Graphs, CoT–TAG) evaluated in Section 5. Strategies differ in how much supervision or explicit reasoning they include; modalities differ in whether the model receives natural language text, a structured graph, or both.

Each strategy is paired with one input modality:

Text : the narrative passage only,

Graphs : a serialized causal graph representing event dependencies,

TAG : both the passage and graph, concatenated.

Zero-shot prompting (Zero) In the Zero track, the model receives task instructions, the input modality (Text, Graphs, or TAG), and the target yes/no question—without demonstrations or reasoning traces. This setting tests whether an LLM can reason directly from the input without prior examples. For instance, using only the Text portion of Figure 1, the model must decide whether "gathered" occurred while the organizers made a statement.

Few-shot prompting (Few) Few prompts add three in-context demonstrations that match the target configuration. Text-only prompts show how to answer using narrative context; graph-based prompts illustrate how causal structure maps to a yes/no label. TAG prompts include both modalities. This setting provides the model with worked examples aligned to the input type.

Chain-of-thought prompting (CoT) CoT prompts build on Few by requesting an explicit reasoning trace. Demonstrations include step-by-step rationales showing how answers are derived from temporal or causal chains. When the graph is present, traces may reference edges (e.g., `BLOCKS`) or event dependencies. This strategy encourages multi-step inference grounded in structured input. See Appendix A.1 for formatting templates across all nine configurations.

3.4 Causal Graph Integration

Each causal graph G is verbalized into natural language using one sentence per edge—e.g., *"Event A* ENABLES *Event B."* or *"Event C* BLOCKS *Event D."* Sentences are ordered topologically to preserve causal flow and reduce reference distance. Events are described using surface forms from the original passage to ensure clarity and self-containment.

Apart from the presence or absence of the passage, all other aspects of the prompt remain fixed: task instructions, in-context examples (in Few), and reasoning traces (in CoT) follow a shared scaffold across modalities. This design isolates the effect of graph structure while controlling for phrasing, format, and token budget.

Examples of full Text, Graphs and TAG prompts for each strategy track appear in Appendix A.1.

3.5 Model Families and Setup

We evaluate three instruction-tuned large language model (LLM) families across the full 3×3 TAG-EQA prompt matrix:

- **T5-XXL** (Google): 11B encoder–decoder model, pretrained with UL2 and fine-tuned on diverse instructions.
- **Qwen-32B (QwQ)** (Alibaba): 32B multilingual decoder trained with chat and instruction tuning.
- **GPT-3.5-Turbo** and **GPT-4o** (OpenAI): proprietary decoder-only models; GPT-3.5 is used for Zero and Few, while GPT-4o is reserved for CoT evaluation on a smaller subset due to cost.

All models use greedy decoding (temperature = 0). Inputs are truncated to model-specific context limits (T5: 1k, Qwen: 2k, GPT: 16k), with graph content prioritized over passage if needed. CoT answers are extracted via regex targeting "Therefore, the final answer is: <yes/no>".

T5 and Qwen are evaluated on both Full and Small subsets; GPT-3.5 runs Zero/Few on Full, and GPT-4o runs CoT on Small due to API constraints.

4 Evaluation

We evaluate **TAG-EQA** using binary classification accuracy: the percentage of questions answered correctly as "yes" or "no." Each model is tested across all nine configurations—three prompting strategies (Zero, Few, CoT) × three input modalities (Text, Graphs, TAG).

For CoT prompts, we extract the final answer using a regex targeting phrases like "Therefore, the final answer is: yes." If absent, we fall back to the first standalone yes/no token[3]. This ensures consistent evaluation across models with variable output formats.

We report results on both the full TORQUESTRA test set (Full, 477K examples) and a 1,024-instance Small subset used for low-resource and cost-sensitive runs (GPT-4o).

To analyze how structure and reasoning affect performance across reasoning types, we group questions into thirteen semantically grounded clusters derived from TORQUESTRA annotations. These extend the original eight-category taxonomy to include finer-grained types such as *positive*, *negative*, *existential*, and *counterfactual*. Accuracy is reported per cluster and per configuration.

See Appendix A.5 for full cluster definitions and results.

5 Results

We evaluate how prompting strategy and input modality affect event-based QA performance across three instruction-tuned LLMs: T5-XXL, Qwen-32B (QwQ), and GPT models (GPT-3.5 and GPT-4o). Each model is tested under nine prompting configurations (Zero/Few/CoT × Text/Graphs/TAG). T5 and Qwen are evaluated on both the full TORQUESTRA test set (Full) and a 1,024-example subset (Small). GPT-3.5 and GPT-4o are evaluated only on the Small subset: GPT-3.5 for Zero and Few (non-reasoning), and GPT-4o for CoT (reasoning), due to API cost and throughput constraints.

Across models, Few-shot prompting consistently outperforms Zero-shot in Text-only settings. CoT prompting yields mixed results: QwQ achieves the highest overall accuracy (74.8%) with TAG-CoT, while T5 performs best with Few-Text. For T5, accuracy drops when CoT is combined with structured input, suggesting difficulty integrating reasoning traces and graph content.

Graphs inputs significantly enhance zero-shot and CoT performance for QwQ, sometimes outperforming TAG inputs. However, modality fusion does not always help: TAG configurations often underperform compared to single-modality prompts, particularly for T5. GPT results remain relatively flat across input types, with Zero-Text (58.7%) performing best for GPT-3.5, and modest gains from CoT in GPT-4o.

These findings highlight the importance of model-aware prompt design: performance gains depend not just on adding structure or reasoning, but on whether a given model can effectively integrate them.

---

[3] Regex: `[Tt]herefore,.*answer is: (yes|no)`

Model	Dataset	Zero	Few	CoT
T5	Full	54.08	58.49	55.21
	Small	52.64	59.47	55.96
QwQ	Full	66.78	70.21	65.77
	Small	68.03	78.32	73.70
GPT	Full	-	-	-
	Small	58.65	52.73	72.28

Table 2: **Prompt-Type Accuracy (%) Comparison on Text–Only Input.** Each model is evaluated on the Full and Small TORQUESTRA subsets. Few–shot prompting consistently outperforms Zero–shot on both scales. CoT shows limited gains on Full, but outperforms Few on Small for QwQ and GPT-4o. GPT results are based on Small only due to cost constraints.

5.1 Does reasoning (CoT) improve performance over Zero or Few-shot using just text?

We begin by comparing Zero, Few, and CoT prompting under Text–only inputs. As shown in Table 2, Few consistently outperforms Zero across models and data sizes. For example, T5 improves from 54.1% to 58.5% on Full, and QwQ improves from 66.8% to 70.2%.

CoT prompting shows mixed effects in the absence of graph input. On the Full set, it underperforms Few for both T5 and QwQ. However, on the Small subset, CoT provides noticeable gains: QwQ improves from 70.2% to 73.7%, and GPT-4o achieves 72.3%, outperforming GPT-3.5's Few score of 52.7%.

These results suggest that chain-of-thought reasoning can help in low-data settings or with models tuned for step-by-step reasoning, such as GPT-4o. Still, Few remains the most reliable strategy when using plain text alone—especially on larger test sets. GPT results are limited to the Small subset: Zero and Few use GPT-3.5, while CoT uses GPT-4o.

5.2 Are Graphs helpful when used alone or combined with Text?

We evaluate the effect of input modality—Text, Graphs, and TAG—under both Zero and Few prompting.

As shown in Table 3a, Graphs–only inputs consistently outperform Text–only across models. For instance, QwQ improves from 66.8% (Text) to 78.8% (Graphs), and T5 gains from 54.1% to 58.0%. Combining Text and Graphs in a TAG prompt further improves performance for QwQ (74.5%) but reduces accuracy for T5 (52.6%). On the Small subset, GPT shows limited variation across modalities – ranging from 56.8% to 58.8% – indicating relative insensitivity to structured input in zero-shot settings. Overall, these results suggest that causal graphs substantially aid zero-shot inference, but modality fusion (Text+Graph) can introduce conflicts depending on the model.

Table 3b shows that Few–shot prompting generally boosts absolute performance compared to Zero. For example, QwQ achieves its highest score (79.4%) with TAG, confirming that demonstrations and graph input are complementary. GPT gains from Graph input (62.9%) compared to Text-only (52.7%), while T5 shows limited or negative gains from structure, dropping from 59.5% (Text) to 50.8% (TAG). These results suggest that few-shot demonstrations amplify the utility of structured graphs for models like QwQ, and GPT, but highlight integration challenges for T5.

5.3 When reasoning is explicitly used, does adding a Graphs help or hurt?

We now examine the effect of input modality under CoT prompting. As shown in Table 4, Graphs–only inputs improve performance for models capable of leveraging structured representations. QwQ achieves its highest accuracy (74.8%) with TAG, while also showing strong performance with Graphs–only input (72.7%).

T5 shows modest gains from Graphs input: on Full, accuracy rises from 55.2% (Text) to 56.9% (Graphs), but drops to 50.4% with TAG, suggesting that reasoning traces may conflict with multimodal inputs for models not tuned for integration. This trend persists on the Small subset.

GPT, evaluated only on Small, shows a slight drop in performance with TAG (70.6%) compared to Text–only input (72.3%), while Graphs–only input yields comparable performance (71.1%). This suggests that GPT–4o does not consistently benefit from structured input when combined with reasoning traces in zero-shot settings.

Overall, these results suggest that graph-augmented reasoning is most effective when the model can exploit structure natively–QwQ benefits most—while other models struggle to integrate multiple information sources effectively under CoT prompting.

Model	Dataset	Text	Graphs	TAG
T5	Full	54.08	57.96	52.58
	Small	52.64	58.89	52.50
QwQ	Full	66.78	78.77	74.48
	Small	68.03	68.09	67.77
GPT	Full	-	-	-
	Small	58.65	56.84	58.79

(a) Zero–shot prompting.

Model	Dataset	Text	Graphs	TAG
T5	Full	58.49	57.54	51.87
	Small	59.47	57.32	50.76
QwQ	Full	70.21	70.48	79.37
	Small	78.32	70.51	78.10
GPT	Full	-	-	-
	Small	52.73	62.99	59.28

(b) Few–shot prompting.

Table 3: **Input Modality Accuracy (%) Comparison. (a)** Zero-shot results: Graphs–only inputs outperform Text–only for most models, with QwQ showing the largest gains. **(b)** Few-shot results: Demonstrations improve overall accuracy, and combining graphs with examples (TAG) is especially effective for QwQ, and GPT, while T5 struggles with multimodal integration.

Model	Dataset	Text	Graphs	TAG
T5	Full	55.21	56.85	50.35
	Small	55.96	56.74	49.56
QwQ	Full	65.77	72.68	74.75
	Small	73.70	71.55	72.05
GPT	Full	-	-	-
	Small	72.28	71.07	70.61

Table 4: **Input Modality Accuracy (%) Comparison in CoT Prompting.** Each model is evaluated using CoT prompting on the TORQUESTRA dataset. T5 and QwQ show modest to strong gains with Graphs inputs. QwQ performs best with combined inputs (TAG), while GPT-4o shows minimal benefit from multimodal prompts. GPT is evaluated only on the Small subset due to API constraints.

Model	Dataset	Best Config	Accuracy%
T5	Full	Few + Text	58.49
	Small	Few + Text	59.47
QwQ	Full	Few + TAG	79.37
	Small	Few + Text	78.32
GPT	Full	-	-
	Small	Zero + Text	72.28

Table 5: **Best-Prompting Configuration per Model.** Top-performing strategy and input modality for each model on the Full and Small TORQUESTRA subsets. GPT results are based on the Small set only due to API cost constraints.

5.4 Which prompting strategy works best for each model?

To better understand model-specific behavior, we report each model's highest-scoring configuration across all nine prompt types (Zero/Few/CoT × Text/Graphs/TAG) for both the Full and Small TORQUESTRA subsets (Table 5). Each entry reflects the optimal combination of prompting strategy and input modality at a given data scale.

QwQ achieves the highest overall accuracy (79.4%) on Full with Few+TAG, showing strong ability to integrate demonstrations and graph input. On Small, it performs best with Few+Text, indicating that graph augmentation is less beneficial under data constraints.

T5 reaches its top accuracy with Few+Text on both subsets (58.5% and 59.5%), showing a clear preference for demonstrations alone. Performance declines when graph input or reasoning traces are included, consistent with earlier observations.

GPT, evaluated only on Small, performs best with Zero+Text (72.3%), suggesting that neither examples nor reasoning traces help much in this setup.

Overall, effective prompting varies by model and scale: structure and reasoning help only when the model can integrate them meaningfully.

5.5 Do certain question types benefit more from reasoning or graphs?

We evaluate model accuracy across thirteen question types derived from TORQUESTRA annotations, extending the original eight clusters (see Appendix A.6 for details). Figure 3 shows accuracy under TAG input—combined text and graph—across Zero-shot, Few-shot, and Graph-CoT prompting. **QwQ and GPT perform best in causal and temporal categories** such as *causal*, *past*, *positive*, and *temporal_conflict*, particularly with CoT prompting. Structured input and reasoning traces appear to help these models handle abstract event relationships.

QwQ and GPT perform best on structured categories—such as *causal*, *past*, *positive*, and *temporal_conflict*—especially when using CoT prompting. Structured input and step-by-step reasoning appear to help these models capture abstract event relationships.

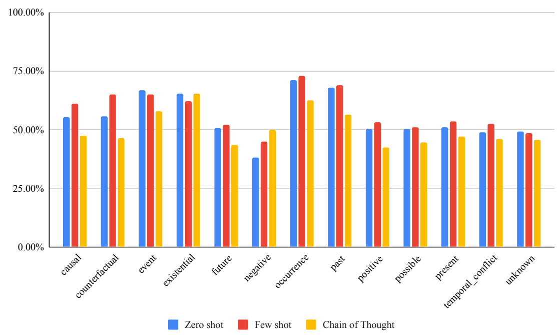
T5 performs best with Few-shot prompting, but its performance drops with Graph-CoT on speculative or underspecified types like *possible*, *present*, and *unknown*, suggesting difficulty integrating structure and reasoning.

Appendix Figures 4 and 5 show that Text prompts gain from Few-shot examples but struggle with relational types, while Graphs prompts provide stronger performance for QwQ in categories like *causal*, *past*, and *event*.

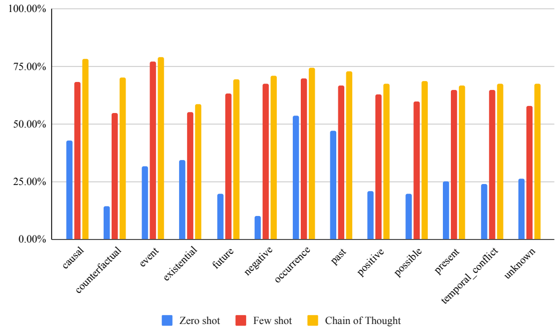
Overall, structured prompting benefits causal and temporal reasoning, with QwQ and GPT showing the strongest gains from graph-augmented CoT. Ambiguous or speculative questions remain difficult across models.

6 Conclusion

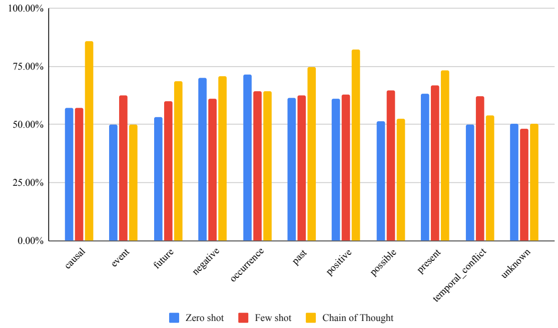
We introduced TAG-EQA, a systematic framework for evaluating event-based question answering (QA) in large language models (LLMs) using structured causal graphs and reasoning-driven prompting. Our experiments covered nine prompting configurations—three strategies (Zero, Few, CoT) crossed with three input modalities (Text, Graphs, TAG)—evaluated on three instruction-tuned LLMs: T5, Qwen (QwQ), and GPT models. Causal graphs consistently improved accuracy on event-centric questions, particularly for relational categories such as *causal*, *past*, and *temporal_conflict*. QwQ achieved the highest overall performance when combining structure and reasoning (TAG+CoT), while T5 performed best with Few+Text and showed limited gains from structured input. GPT models, evaluated only on a smaller subset, showed moderate benefits from CoT prompting but little sensitivity to input modality. Ambiguous or underspecified categories—such as *possible* and *unknown*—remained challenging across models and prompting styles. These findings highlight both the strengths and limitations of using structured causal input to guide reasoning in LLMs.



(a) T5 under TAG: *Few—Text* is strongest overall; *Graph—CoT* tends to underperform on speculative or underspecified types (*possible*, *present*, *unknown*).



(b) QwQ under TAG: *Graph—CoT* generally leads on structured types (*causal*, *past*, *temporal_conflict*); *Few—Text* remains competitive elsewhere.



(c) GPT under TAG: *Graph—CoT* improves relational categories (e.g., *causal*, *temporal_conflict*); strategy gaps narrow on underspecified types (*possible*, *unknown*).

Figure 3: **Cluster-wise accuracy under the TAG configuration.** Bars denote *Zero—Text* (blue), *Few—Text* (red), and *CoT* with TAG (yellow) across thirteen question types. Subfigures (a–c) report T5, QwQ, and GPT respectively. Text–only and Graph–only cluster results appear in Appendix Figures 4 and 5.

7 Limitations and Future Work

Our evaluation relies on expert-annotated causal graphs from TORQUESTRA, which provide clean structure but do not reflect the sparsity or noise of automatically induced graphs. The reported numbers should therefore be interpreted as an *upper bound* on the benefit of structured input. Prompt construction is also manually designed—including example selection and reasoning trace format—which may limit generalization to new domains without automation. Performance further varies across models: QwQ benefits most from graph-augmented CoT prompting, whereas T5 and GPT show more modest or inconsistent gains. Due to API cost constraints, GPT models are evaluated only on the Small subset—a full-scale CoT run with GPT-4o would exceed $950.[4] Lastly, our binary QA task simplifies causal reasoning and does not capture the complexity of multi-hop inference or generative outputs.

Beyond these constraints, our study is limited to three instruction-tuned LLM families (T5, QwQ, GPT); other architectures may respond differently to structured prompts. We also restrict evaluation to TORQUESTRA, leaving extensions to broader narrative QA datasets (e.g., NarrativeQA, MCTest) for future work. Finally, while we report average prompt lengths, a systematic study of context budget and scaling effects remains open.

Future directions include automated graph construction, robustness to noisy or incomplete graphs, and adaptive graph selection to filter only edges relevant to a query. Extending TAG-EQA with dynamic reasoning traces, instruction tuning for graph-structured CoT prompting, and applications to generative or interactive tasks—such as story simulation, causal forecasting, or decision support—offers promising next steps for leveraging structured knowledge in real-world applications.

Acknowledgments

We thank the reviewers for their detailed comments and suggestions. Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by DARPA for the SciFy program under agreement number HR00112520301. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA or the U.S. Government.

References

Mazal Bethany, Emet Bethany, Brandon Wherry, Cho-Yu Chiang, Nishant Vishwamitra, Anthony Rios, and Peyman Najafirad. 2024. Enhancing event reasoning in large language models through instruction fine-tuning with semantic causal graphs. *arXiv preprint arXiv:2409.00209*.

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

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797. Association for Computational Linguistics.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. 2024. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670.

Jesse Dunietz, Sam Thomson, Chris Dyer, and Noah A. Smith. 2020. An interpretable, lexicalized model for implicit event causality. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1713.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2463–2473.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning to predict from textual data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 613–622.

Michael Regan, Jena D. Hwang, Keisuke Sakaguchi, and James Pustejovsky. 2023. Causal schema induction for knowledge discovery. *arXiv preprint arXiv:2303.15381*.

Kaushik Roy, Alessandro Oltramari, Yuxin Zi, Chathurangi Shyalika, Vignesh Narayanan, and Amit Sheth. 2024. Causal event graph-guided language-based spatiotemporal question answering. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 227–233.

---

[4] See Appendix A.3 for detailed cost calculations.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Khurram Yamin, Shantanu Gupta, Gaurav R Ghosal, Zachary C Lipton, and Bryan Wilder. 2024. Failure modes of llms for causal reasoning on narratives. *arXiv preprint arXiv:2410.23884*.

Yao Zhang, Cheng Deng, Wayne Xin Zhao, Bing Qin, and Ting Liu. 2023. Automatic chain-of-thought prompting in large language models. *arXiv preprint arXiv:2302.00923*.

A Appendix

This appendix provides additional implementation details, example prompts, and full evaluation results referenced in the main paper. We include: (1) prompt format illustrations, (2) input component breakdowns, (3) per-model prompting results, (4) API cost estimates, and (5) expanded cluster definitions and analysis.

A.1 Prompt Format Examples

We show two full prompt examples in the CoT setting: one using only a causal graph (Graphs) and one using both the passage and graph (TAG). These correspond to the instance in Figure 1.

A.1.1 Graphs – CoT Prompt

```
### Instruction ###
You are provided with a causal graph and
examples showing how to answer. Use only
the graph and answer “yes” or “no” only.

### Graph ###
The event "riot police deployed" blocks
the event "protest rally".
The event "political opposition" enables
the event "political opposition called
rally".
The event "political opposition called
rally" enables the event "protest
rally".
The event "music" enables the event
"draws many people to festival".
The event "dancing" enables the event
"draws many people to festival".
The event "speeches" enables the event
"draws many people to festival".

### Examples ###
Question: Did "protest rally" happen
after "riot police deployed"?
Answer: no
Question: Did "music" cause "draws many
people"?
Answer: yes

### Question ###
Did "gathered" happen while the
organizers made a statement?
### Answer ###
```

Modality	Input	Zero	Few	CoT
Text	Text	✓	✓	✓
	Graph			
	Examples		✓	✓
Graphs	Text			
	Graph	✓	✓	✓
	Examples		✓	✓
TAG	Text	✓	✓	✓
	Graph	✓	✓	✓
	Examples		✓	✓

Table 6: Components used in each prompt configuration.

A.1.2 TAG – CoT Prompt

```
### Instruction ###
You are provided with text, a causal
graph, and examples showing how to
answer. Integrate both and answer “yes”
or “no” only.

### Text ###
Organizers state the two days of music,
dancing, and speeches is expected to
draw some two million people. But as
supporters gathered in the north, riot
police deployed in Lagos to break up a
protest rally called by the political
opposition.

### Graph ###
[Same graph as above]

### Examples ###
[Same examples as above]

### Question ###
Did "gathered" happen while the
organizers made a statement?
### Answer ###
```

A.2 Prompt Component Matrix

Table 6 summarizes which components appear in each of the nine prompting configurations used in TAG-EQA.

A.3 API Cost Estimate

We compute cost estimates for GPT-3.5 and GPT-4o using OpenAI’s May 2025 pricing. Table 7 shows that a full CoT evaluation with GPT-4o would exceed $950, so we restrict GPT results to the Small subset.

Model	Prompt Type	Total Tokens	Cost (USD)
GPT-3.5	Zero-shot	9.7M in / 53k out	$4.95
GPT-3.5	Few-shot	12.6M in / 53k out	$6.36
GPT-4o	CoT	21.1M in / 212M out	$957.16

Table 7: Estimated cost to run GPT models on Full dataset.

A.4 Per-Model Prompting Results

We report accuracy for each model across all 3 × 3 prompting configurations. These tables complement Section 5.4 and clarify which strategies and modalities are most effective for different architectures.

T5. Performs best with Few+Text, but degrades when structure or reasoning is added.

Prompt Type	Text	Graphs	TAG
Zero	54.1	58.0	52.6
Few	58.5	57.5	51.9
CoT	55.2	56.9	50.4

Table 8: T5 accuracy across all strategies and modalities.

QwQ. Excels with TAG+Few and TAG+CoT. Gains are consistent across most settings.

Prompt Type	Text	Graphs	TAG
Zero	66.8	66.8	74.5
Few	70.2	70.5	79.4
CoT	65.8	72.7	74.8

Table 9: QwQ accuracy across all strategies and modalities.

GPT. Best performance under CoT (GPT-4o). GPT-3.5 shows smaller gains and flat modality sensitivity.

Prompt Type	Text	Graphs	TAG
Zero	58.7	56.8	58.8
Few	52.7	63.0	59.3
CoT	72.3	71.1	70.6

Table 10: GPT accuracy across all strategies and modalities.

Note: All GPT results are reported on the Small subset due to API cost constraints.

A.5 Expanded Cluster Definitions

We extend TORQUESTRA's original eight cluster categories into thirteen to better capture event-centric reasoning. Table 11 aligns our expanded taxonomy with the original groups.

A.6 Cluster-Based Accuracy Analysis

We present accuracy trends by question category across all prompting configurations.

Expanded Category	Original Cluster
causal	causal
counterfactual	causal (extended)
event	event
existential	event (subtype)
future	future
negative	event (negative polarity)
occurrence	event / temporal
past	past
positive	event (positive polarity)
possible	possible
present	present
temporal_conflict	temporal_conflict
unknown	unknown

Table 11: Expanded category mapping.

T5: Best with Few+Text on most clusters. Accuracy drops with Graphs+CoT.

QwQ: Excels with TAG+CoT. Leads in most structured and relational categories.

GPT (3.5/4o): CoT (GPT-4o) performs best across categories like *causal* and *past*; GPT-3.5 (Zero/Few) is stable but less sensitive to modality.

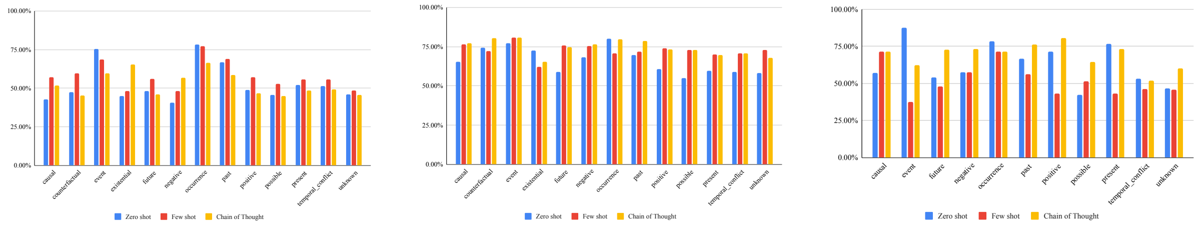


Figure 4: **Cluster-wise Accuracy by Model and Prompting Strategy.** Accuracy across thirteen question categories for each model (T5, QwQ, GPT) under three prompting strategies: Zero–Text (blue), Few–Text (red), and CoT with TAG input (yellow). QwQ and GPT benefit most from graph-augmented CoT prompting on structured categories such as *causal*, *past*, and *temporal_conflict*. T5 performs best with Few–shot but struggles to integrate structure and reasoning. All models show weaker performance on underspecified or speculative categories like *possible* and *unknown*.

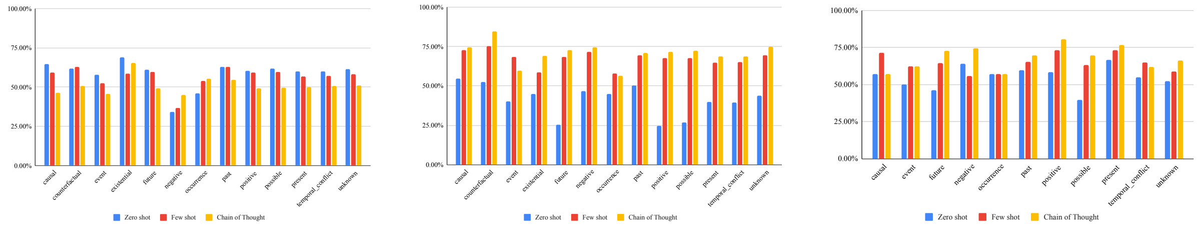


Figure 5: **Cluster-wise Accuracy by Model and Prompting Strategy.** Accuracy across thirteen question categories for each model (T5, QwQ, GPT) under three prompting strategies: Zero–Text (blue), Few–Text (red), and CoT with TAG input (yellow). QwQ and GPT benefit most from graph-augmented CoT prompting on structured categories such as *causal*, *past*, and *temporal_conflict*. T5 performs best with Few–shot but struggles to integrate structure and reasoning. All models show weaker performance on underspecified or speculative categories like *possible* and *unknown*.

DisCoCLIP: A Distributional Compositional Tensor Network Encoder for Vision-Language Understanding

Kin Ian Lo and **Hala Hawashin** and **Mina Abbaszadeh**
Tilen Limback-Stokin and **Hadi Wazni** and **Mehrnoosh Sadrzadeh**
University College London

Abstract

Recent vision–language models excel at large-scale image–text alignment but often neglect the compositional structure of language, leading to failures on tasks that hinge on word order and predicate–argument structure. We introduce DisCoCLIP, a multimodal encoder that combines a frozen CLIP vision transformer with a novel tensor network text encoder that explicitly encodes syntactic structure. Sentences are parsed with a Combinatory Categorial Grammar parser to yield distributional word tensors whose contractions mirror the sentence's grammatical derivation. To keep the model efficient, high-order tensors are factorized with tensor decompositions, reducing parameter count from tens of millions to under one million. Trained end-to-end with a self-supervised contrastive loss, DisCoCLIP markedly improves sensitivity to verb semantics and word order: it raises CLIP's SVO-Probes verb accuracy from 77.6% to 82.4%, boosts ARO attribution and relation scores by over 9% and 4%, and achieves 93.7% on a newly introduced SVO-Swap benchmark. These results demonstrate that embedding explicit linguistic structure via tensor networks yields interpretable, parameter-efficient representations that substantially improve compositional reasoning in vision–language tasks.

1 Introduction

Vision-language understanding is a key challenge in AI, with applications to image captioning and multimodal retrieval. Models like OpenAI's CLIP (Radford et al., 2021) have shown that large-scale joint embeddings can effectively connect visual and textual data. However, these models mainly rely on Transformer architectures with dense attention, which may overlook the linguistic structure. For instance, recent evaluations of CLIP-like models show that they often ignore word order, acting like bags-of-words (Thrush et al., 2022; Jiang et al., 2024; Li et al., 2024). The Attribution, Relation and Order (ARO) benchmark (Yuksekgonul et al., 2023) checks if they are able to understand the correct word order. Similarly, the SVO-probes benchmark (Hendricks and Nematzadeh, 2021) tests if these models mainly focus on nouns, or are also able to recognise verbs. Both of these issues have been common challenges for vision-language models.

It has been argued that these challenges stem from CLIP-like models being trained on web-sourced image-caption pairs, where captions (often alt-texts) frequently ignore word order and verb usage. As a result, their contrastive learning is not sensitive to linguistic structure (Yuksekgonul et al., 2023). While training with *hard negatives* could address this, such samples are costly to source. Instead, we introduce **DisCoCLIP**, the first model for vision and language with a text encoder that fully incorporates the compositional linguistic structure of text with the distributions of the words therein. To achieve this, we represent sentences as tensor networks, where each word is encoded as a tensor and interactions between words are captured through a series of tensor contractions.

The advantages of using a tensor network text encoder are twofold. First, it enables explicit encoding of both syntactic structure and statistical semantic information, making the resulting text representations more interpretable than those produced by transformer-based encoders. Second, tensor network decompositions can dramatically reduce the number of parameters required, allowing for efficient modelling of high-order interactions without incurring exponential growth in tensor size. Tensor networks are widely used in quantum machine learning to capture higher order data correlations (Biamonte et al., 2017; Schuld et al., 2015; Stoudenmire and Schwab, 2016; Cichocki et al., 2016). Their use in vision-language tasks might lead to further advantages coming from the quan-

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 316–327
November 8-9, 2025 ©2025 Association for Computational Linguistics

tum world.

DisCoCLIP was evaluated on two existing benchmarks on compositional capability: SVO-Probes and ARO, as well as on a new SVO-Swap benchmark created by swapping subjects and objects. We compare the performance of **DisCoCLIP** with CLIP, OpenCLIP (Ilharco et al., 2021) and BLIP (Li et al., 2022) on these benchmarks.

DisCoCLIP outperforms CLIP and OpenCLIP on verb understanding by 4.82% and 1.01%. It also outperforms CLIP in overall performance by 1.3%, but falls behind OpenCLIP and BLIP by 2.05% and 7.9%. On SVO-Swap, it achieves an accuracy of 93.68% outperforming all three of CLIP, OpenCLIP and BLIP by a large margin ($30.52\% - 57.04\%$). On ARO-Relation, again it outperforms all three of the CLIP models by $4.28\% - 5.1\%$, in ARO-Attribution, it outperforms CLIP and OpenCLIP by 9.01% and 10.88%, but falls behind BLIP by 8.45%.

In summary, DisCoCLIP achieves comparable performance to transformer-based models with orders of magnitude fewer parameters. The use of tensor decomposition enables efficient representation and computation, making our model more parameter-efficient and potentially more robust when training data is limited. To our knowledge, it is the first time that the theory of tensor networks has been used to model the structure of language or used in vision-language tasks. Our work provides a new witness for the applications of tensor networks to machine learning and further showcases the advantage of using them.

2 Related Work

Several approaches have been proposed to address these challenges in vision-language models. Some incorporate aspects of linguistic structure (Jiang et al., 2024), others introduce hard negatives (Li et al., 2024), and some incentivize learning by explicitly rewarding the model for capturing linguistic elements such as adjectives and verbs (Thrush et al., 2022).

Tensor networks were introduced to make the numerical treatment of many-body quantum states feasible by exploiting their internal structure (White, 1992). Such states naturally live in exponentially large tensor-product spaces, which are difficult to handle directly. A tensor network circumvents this by factorizing a single, high-order tensor into a set of lower-order tensors, whose indices are glued together by contraction operations. In a *Tensor Train* (also known as a Matrix Product State, or MPS), these tensors are arranged in a strictly one-dimensional sequence, with each tensor contracted only to its immediate predecessor and successor through shared *bond* indices; by contrast, a Tree Tensor Network connects tensors in a branching, hierarchical structure. Tensor networks have found applications outside physics, especially in machine learning where they are used for sequence modelling (Harvey et al., 2025), optimizing the computations of neural networks (Ahromi and Orús, 2024; Novikov et al., 2015), and in general any large-scale optimization problem (Cichocki et al., 2017), such as latent feature extraction (Stoudenmire, 2018) and security (Aizpurua et al., 2025). Their decomposition methods have been tested on image classification tasks (Roberts et al., 2019; Rao et al., 2020; Serafini and d'Avila Garcez, 2017), word statistics, and document retrieval from large corpora of text (Miller et al., 2021; Zhang et al., 2019; Liu et al., 2005; Bouchard et al., 2015).

Tensors and the contraction operation between them were also used in a model of meaning known as "compositional distributional semantics" (Baroni and Zamparelli, 2010; Maillard et al., 2014; Grefenstette and Sadrzadeh, 2011; Yeung and Kartsaklis, 2021). In this model, the meaning of each word is either a vector or a higher-order tensor. The orders of the tensors are determined by the grammatical roles of words. Meanings of nouns are vectors, where as meanings of words with functional roles such as adjectives and verbs are matrices and cubes. **DisCoCLIP** is inspired by compositional distributional semantics and the theory of tensor networks. We denote the meaning of a piece of text by a tensor network. In this tensor network, the tensors encode meanings of words, the layout of the tensor network represents the syntactic structure of the sentence. Other tensor network layouts are used as baselines to test how useful is encoding less structure, such as word order and bags-of-words.

Another key novelty of our model is that it extends compositional distributional semantics to a multimodal setting. Previous multimodal adaptations include (Lewis et al., 2024) for compositional concept learning, (Nazir and Sadrzadeh, 2024) for audio-text retrieval, and (Wazni et al., 2024) for verb understanding in CLIP. However, **DisCoCLIP** differs from these approaches in two important ways. First, our model is more general: It handles sentences of arbitrary syntactic structure,

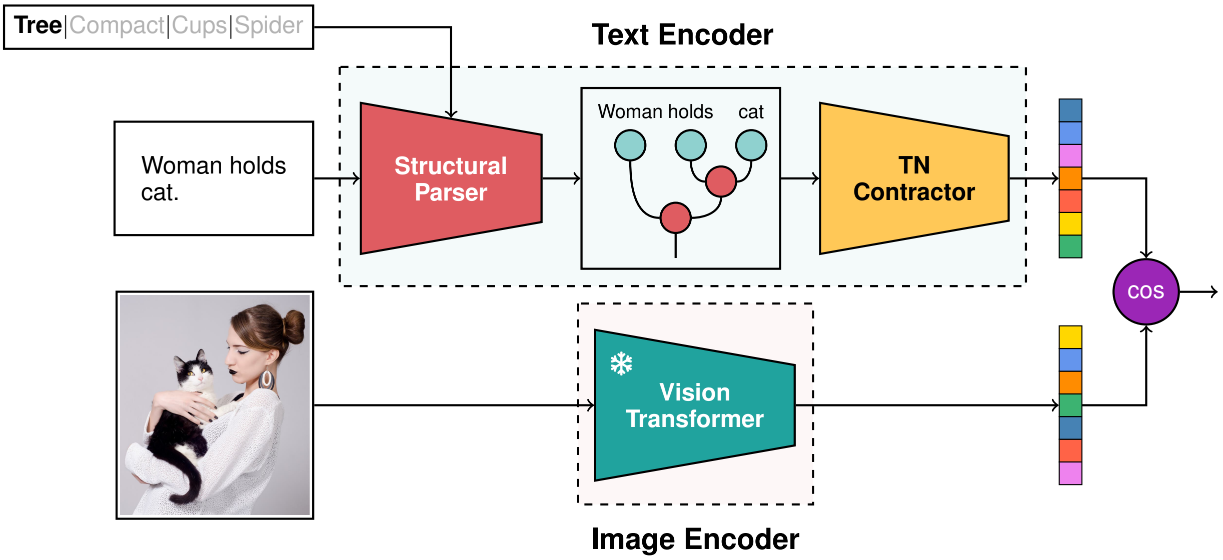


Figure 1: An illustration of the architecture of DisCoCLIP, which consists of a text encoder based on a structure-informed tensor network of words, and a vision encoder based on a Vison Transformer (ViT). The Structural Parser converts the input text into a tensor network, based on the chosen structure which could be any of the four types: **Tree**, **Compact**, **Cups** or **Spider**. The tensor network is then contracted by the Tensor Network Contractor, which computes an optimal contracting order to obtain a single vector representing the meaning of the input text. The input image is processed by Vision Transformer (ViT) to obtain a vector representation of the image. The text and image vectors are then used to compute a similarity score, which is used for training the model and for downstream evaluation.

whereas prior work typically focuses on specific constructions such as subject-verb-object (Lewis et al., 2024; Wazni et al., 2024) or adjective-noun pairs (Nazir and Sadrzadeh, 2024). Second, **DisCoCLIP** features an end-to-end pipeline trained with a single objective function, in contrast to previous methods that require a separate objective for their different text and audio/image model components. This unified approach enables more flexible and scalable multimodal learning.

3 Basics of Tensor Networks

A tensor network is a collection of tensors contracted together to form a new tensor. An order-n tensor T is a multi-dimensional array $T \in \mathbb{R}^{d_1 \times \cdots \times d_n}$, where d_i is the dimension of the i-th index. Elements are denoted by $T_{i_1,\ldots,i_n}$, with each i_k ranging from 0 to d_{k-1}. Scalars, vectors, and matrices are tensors of order 0, 1, and 2, respectively.

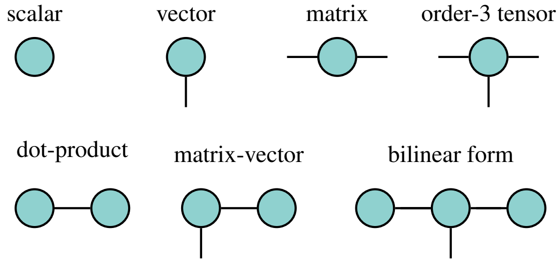


Figure 2: Graphical representation of tensor networks. A tensor is depicted as a node with one edge for each index of the tensor. For example a scalar has no edge, a vector has one edge, a matrix has two edges and an order-3 tensor has 3 edges. An edge of a node can be connected to another edge of another node, forming a *contraction*, which is a generalised form of matrix multiplication.

Tensor contractions. Tensors can be multiplied together by contracting over a shared index, which generalizes matrix multiplication. For example, given two tensors $A \in \mathbb{R}^{d_1 \times d_2}$ and $B \in \mathbb{R}^{d_2 \times d_3}$, their contraction over the second index yields a new tensor $C \in \mathbb{R}^{d_1 \times d_3}$:

$$C_{i_1,j_1} = \sum_{k=1}^{d_2} A_{i_1,k}\, B_{k,j_1}$$

This operation extends naturally to higher-order tensors by summing over any shared index.

$$C_{i_1,\ldots,i_p,j_1,\ldots,j_q} = \sum_{k_1,\ldots,k_r} A_{i_1,\ldots,i_p,k_1,\ldots,k_r} \times B_{k_1,\ldots,k_r,j_1,\ldots,j_q}$$

where the indices $k_1, \ldots, k_r$ are summed over, representing the contracted dimensions shared by A and B. This operation generalizes matrix multiplication and inner product to higher-order tensors.

Graphical representation. Tensor contractions involving multiple tensors can be difficult to reason about. The graphical representation of tensors, as shown in Figure 2, provides a more intuitive way of visualizing them. In this representation, tensors are depicted as nodes and their indices as edges, with edges common to two tensors indicating a contraction.

Tensor decomposition. As the number of parameters grows exponentially with the tensor order, computing with them becomes costly. Tensor networks were originally introduced to efficiently represent high-order tensors by decomposing them into a network of lower-order tensors. The number of parameters of an order-n tensor $T \in \mathbb{R}^{d_0 \times d_1 \times d_2 \times \cdots \times d_n}$ is given by the product of its dimensions, $d_0 d_1 d_2 \cdots d_n$.

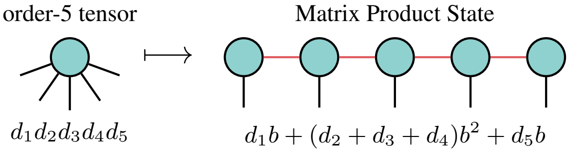


Figure 3: The decomposition of an order-5 tensor into a Matrix Product State (MPS). The red edges are called the bonds and their dimension is called the *bond dimension* b. Below the tensors, we show the formulas for the number of parameters required to represent the full order-5 tensor (bottom left) and its MPS decomposition (bottom right).

In many practical scenarios, representing a high-order tensor with all of its exponentially many parameters is unnecessary. Instead, the tensor can often be efficiently approximated or even exactly represented by decomposing it into a network of lower-order tensors. This decomposition, called a tensor network, greatly reduces the number of parameters and enables scalable computation.

A canonical example is the ground state of a quantum many-body system, which can be efficiently represented by a Matrix Product State (MPS) (Fannes et al., 1992), also known as a Tensor Train. An MPS expresses a high-order tensor as a sequence (or "train") of lower-order tensors connected by contracted indices, as illustrated in Figure 3. The contractions between neighboring tensors are called *bonds* and their dimensions are called *bond dimensions*. The dimension of each bond index is the *bond dimension* b, which controls the expressiveness and parameter count of the MPS. The total number of parameters in the MPS is

$$(d_1 + d_n) b + \sum_{k=2}^{n-1} d_k b^2,$$

assuming all bond dimensions are equal to b. This is typically much smaller than the $d_1 d_2 \cdots d_n$ parameters required for a full tensor, making MPS an efficient representation for high-order tensors. For the rest of this paper, we will use a uniform dimension denoted by d and a uniform bond dimension denoted by b for simplicity. We denote the number of parameters in an MPS representation of an order-n tensor as

$$\#\text{MPS}(n, d, b) = \begin{cases} d, & n = 1 \\ 2db + (n-2)db^2, & n \geq 2 \end{cases} \tag{1}$$

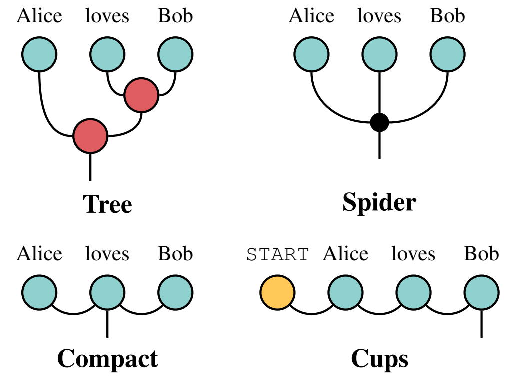


Figure 4: The four types of tensor networks considered in this paper: **Compact** and **Tree** are based on the CCG grammar, **Cups** preserves word order and **Spider** is a bag-of-words model. Each rectangle represents a node in the tensor network. The black dot in Spider is the copy node, which is operationally equivalent to element-wise multiplication.

Other than MPS, other common tensor network decompositions include the Tree Tensor Network (TTN) (Shi et al., 2006), which arranges tensors in a tree structure, and the Projected Entangled Pair State (PEPS) (Verstraete and Cirac, 2004), which arranges tensors in a 2D lattice. These decompositions are useful for different applications and can be adapted to specific data structures.

4 Methodology

Our main contribution is to replace CLIP's Transformer-based text encoder with a tensor network encoder, resulting in a new vision-language model we call **DisCoCLIP**. In DisCoCLIP, the text encoder constructs sentence embeddings using tensor networks that explicitly encode linguistic structure, while the image encoder remains the original CLIP vision transformer. By varying the layout of the tensor network, we can control the level of syntactic and semantic information captured in the text representation.

Given an image-caption pair, **DisCoCLIP** processes them in the following steps (see Figure 1):

1. The sentence is parsed to extract its syntactic structure.
2. A tensor network is constructed based on the parse tree, where each word is represented by a tensor node.
3. The tensor network is contracted to produce a fixed-size vector embedding for the entire sentence.
4. The image embedding is computed using a Vision Transformer (ViT).
5. The text and image embeddings are compared to compute a similarity score, which is used for training the model and for downstream evaluation.

For step 1, we use the state-of-the-art `BobcatParser` (Clark, 2021) from the `Lambeq` library (Kartsaklis et al., 2021) to obtain the Combinatory Categorial Grammar parse trees of the sentences (Ades and Steedman, 1982; Steedman, 1987, 2000) .

Combinatory Categorial Grammar (CCG) is a highly expressive formalism for modeling natural language syntax and semantics. In CCG, each word is assigned a syntactic category that reflects both its grammatical role and its combinatory potential. Categories are either atomic (such as noun phrase NP or sentence S) or functional, where functional categories specify how a word combines with its arguments. Functional types take the form Y/X or $Y\backslash X$, indicating that the word expects an argument of type X to its right (/) or left (\), and yields a result of type Y. For example, adjectives have type NP/NP, intransitive verbs have type $S\backslash NP$, and transitive verbs have type $(S\backslash NP)/NP$.

The combinatory rules of CCG allow for the composition of these categories to cancel out the functional types and yield a sentence S type. The two main rules are forward application (>) and backward application (<):

$$\frac{X/Y \quad Y}{X}> \qquad \frac{Y \quad X\backslash Y}{X}<$$

where X and Y are any CCG types. These rules allow for the composition of words into phrases and sentences, following the syntactic structure of the language. For example, the sequence “Alice loves Bob” can be reduced to a sentence S by first assigning the atomic category NP to both “Alice” and “Bob”, and the functional category $(S\backslash NP)/NP$ to “love” and then applying the forward and backward application rules as follows:

$$\cfrac{\cfrac{Alice}{NP} \quad \cfrac{\cfrac{loves}{(S\backslash NP)/NP} \quad \cfrac{Bob}{NP}}{S\backslash NP}>}{S}<$$

Other CCG rules include forward and backward composition, which are used to combine auxiliary verbs with their arguments, and forward and backward cross-composition, used to combine categories with long distance dependencies such as gapping. Another notable CCG rule is type-raising, which enables specific combinations of categories, e.g. from left to right. This feature helps the CCG align with Psycholinguistic theories. For instance, in English, it will allow categories to combine from left to right and form incremental parses that support theories of human sentence processing.

A distributional compositional (DisCo) semantics has been developed for CCG (Grefenstette and Sadrzadeh, 2011; Yeung and Kartsaklis, 2021; Wijnholds et al., 2020). This semantics assigns to a word w with a CCG category composed of n atomic categories a multilinear map f_w with n arguments

$$f_w : V_1 \times V_2 \times \cdots \times V_n \to V_{n+1}$$

Each V_i is a finite-dimensional vector space over the field of reals $\mathbb{R}$. Equivalently, f_w can be represented by a tensor of in the space

$$f_w \in V_1 \otimes V_2 \otimes \cdots \otimes V_{n+1}$$

Here, each atomic type corresponds to an index of the tensor. For example, a noun with the type NP is assigned a vector (order-1 tensor), while an adjective with the type NP/NP is assigned a linear map that takes a vector and returns a vector, which can be represented as a matrix (order-2 tensor). A transitive verb with the type $(S\backslash NP)/NP$ is assigned a bilinear map that takes two vectors and returns another vector, i.e. a cube (an order-3 tensor), and so on. For the general formulae of these representations, see (Maillard et al., 2014) and (Wijnholds et al., 2020).

Given the CCG parse tree, the word tensors are composed by performing tensor contractions that mirror the syntactic reductions specified by the tree. Each time a combinatory rule (such as forward or backward application) is applied in the parse, the corresponding word tensors are contracted along the appropriate indices. This process recursively combines the tensors according to the grammatical structure, ultimately yielding a single vector representation for the entire sentence. Such semantics was developed in (Maillard et al., 2014; Wijnholds et al., 2020) and leads to the **Compact** tensor network structure.

An alternative semantics presented in (Yeung and Kartsaklis, 2021) assigns to every word a vector and models the grammatical compositions (represented by CCG rules such as forward and backward application) by a shared order-3 tensor. This tensor acts as a universal composition operator of all compositional operators. This approach yields the **Tree** tensor network structure, where the parse tree topology is preserved but all internal nodes use the same composition tensor to combine their child representations.

4.1 Text Encoder Structures

We consider four types of tensor network structures: **Tree**, **Compact**, **Cups**, and **Spider**, as illustrated in Figure 4. Every tensor node in the networks is a trainable parameter, which is learned during the training process.

The **Tree** structure is based on the CCG parse tree of the sentence, where each word is represented as a vector node and an order-3 tensor is used to compose these word nodes to form non-terminal terms in the parse tree.

The **Compact** structure is a variant of the **Tree** structure, where every non-terminal node in the parse tree is absorbed by one of its parents, resulting in a more compact representation where some word nodes become higher-order tensors.

The **Cups** structure is a variant of Tensor Train (or MPS) where each word is an order-2 tensor, connected in a chain to preserve word order. The first word connects to a special `start` node while the last word outputs the sentence embedding.

The **Spider** structure implements a bag-of-words model, where each word is represented as a vector node and all word nodes are contracted through a special *copy node* to produce a single output vector. This copy node, of order n, is a tensor $C \in \mathbb{R}^{d^n}$ defined as

$$C_{i_1,i_2,\dots,i_n} = \begin{cases} 1 & \text{if } i_1 = i_2 = \cdots = i_n, \\ 0 & \text{otherwise.} \end{cases}$$

Contracting $n-1$ indices of the copy node with $n-1$ word vectors yields their element-wise (Hadamard) product, producing a multiplicative bag-of-words sentence embedding.

Parameter count. Each tensor network structure has a different parameter count, determined by the number and order of word tensors and any composition tensors. Let $|V|$ be the vocabulary size. For **Compact**, let $|V^r|$ be the number of words with order-r tensors, and $\#\text{MPS}(r,d,b)$ the parameter count for an order-r MPS (see Eq. (1)). Table 1 summarizes the counts.

Structure	Words	Composition
Tree	$\|V\|d$	$2db + db^2$
Compact	$\sum_r \|V^r\|\#\text{MPS}(r,d,b)$	0
Spider	$\|V\|d$	0
Cups	$\|V\|d^2$	d

Table 1: Number of parameters for each tensor network structure.

5 Contrastive Learning

We train **DisCoCLIP** using contrastive learning, where the image encoder f (frozen CLIP) and the tensor network text encoder g map image-caption pairs (x, y) to embeddings $(\mathbf{x}, \mathbf{y})$. The goal is to bring true (positive) pairs closer and push (negative) mismatched pairs apart in the joint embedding space. For a batch of B positive pairs, all non-matching image-caption combinations in the batch ($B(B-1)$) serve as negatives, following the in-batch negative sampling of CLIP (Radford et al., 2021). These are considered *easy* negatives, as opposed to more challenging, hand-crafted *hard* negatives.

We use the widely adopted InfoNCE loss (van den Oord et al., 2018) to train the model. Given a batch of B image-caption pairs with embeddings $\mathbf{x}_i = f(x_i)$ and $\mathbf{y}_i = g(y_i)$, the InfoNCE loss is

$$\mathcal{L} = -\sum_{i=1}^{B} \log \frac{\exp(s(\mathbf{x}_i, \mathbf{y}_i)/\tau)}{\sum_{j=1}^{B} \exp(s(\mathbf{x}_i, \mathbf{y}_j)/\tau)}, \qquad (2)$$

where τ is a temperature parameter and $s(\mathbf{x}, \mathbf{y})$ is the cosine similarity between the image embedding

$\mathbf{x}$ and the caption embedding $\mathbf{y}$. Here, the numerator measures similarity for positive (matching) pairs ($i = j$), while the denominator includes all pairs in the batch, serving as negatives when $i \neq j$. The loss thus encourages higher similarity for true pairs and lower for mismatched ones.

We evaluate our approach on two key benchmarks for vision-language understanding: SVO-Probes and ARO. The SVO-Probes dataset is designed to test whether models can distinguish fine-grained changes in the image which corresponds to variations in subject, verb, or object. The task is to determine which of the two images correctly matches a given caption. In contrast, the ARO (Attribution, Relation, and Order) dataset assesses a model's ability to correctly compose meanings in a sentence. The task is to determine which of the two captions correctly describes a given image. Together, these datasets provide a comprehensive evaluation of both compositional and structural language understanding in multimodal models.

5.1 SVO-Probes

For SVO-Probes, we prompted the language model `Llama-3.2-3B` (Grattafiori et al., 2024) to correct grammatical and spelling mistakes as the original dataset contained errors from crowdsourced captions. The exact prompt used can be found in the Appendix A. The images in the SVO-Probes dataset were not available for download from the official repository; therefore, we attempted to download them from the Internet using the provided URLs on 2 May 2025. However, many of the URLs were no longer active, and we were only able to download 8,984 images of the total 14,097 images in the dataset, resulting in a reduction of the dataset size from 36,841 to 20,458 entries.

To ensure a reasonable train-test vocabulary overlap, we filtered out entries that contained words that appeared fewer than 50 times in the entire dataset, yielding 8,984 image-caption pairs split 60/20/20 for training, validation, and test, with no image overlap between splits. We also introduced a new dataset: SVO-Swap. This is a set of 95 evaluation pairs created by swapping subjects and objects (when both refer to humans or animals) in SVO-Probes captions.

The SVO-Probes benchmark is divided into three subsets: Subject, Verb, and Object. Each of these corresponds to the specific component of the sentence that differs between the two alternatives. This structure enables a fine-grained evaluation of the model's ability to distinguish changes in the linguistic roles of the words within a caption.

5.2 ARO

The ARO dataset consists of four different subsets: Visual Genome Attribution (VG-A), Visual Genome Relation (VG-R), COCO Order and Flickr Order. The way these subsets are constructed was to first gather a set of positive image-caption pairs, and then apply a certain modification to the captions to form negative captions. In the VG-A subset, positive pairs are chosen to be images with two objects and each gets an attribute. For example *the silver fork and the round plate* contains a fork that is silver, and a plate that is round. The corresponding negative caption would be *the round fork and the silver plate*, where the attributes for the two objects are swapped. For the VG-R subset, positive pairs are images with a relation involving two objects. For example For ARO-Attribution (with 28,748 entries) and ARO-Relation (with 23,937 entries), we used a 70/15/15 split without frequency filtering, as vocabulary overlap was sufficient.

5.3 Training

For each structure (**Tree**, **Compact**, **Spider** and **Cups**), we trained the tensors for 10 epochs, using the `AdamW` optimizer (Loshchilov and Hutter, 2019) with a learning rate of 10^{-3}, a weight decay of 10^{-2} and a batch size of 64. We also experimented with bond dimensions 2, 5, 10, 15 and 20 in the MPS decomposition. Training was performed on an Apple M1 MacBook with 16GB RAM, utilizing the PyTorch Metal Performance Shaders (`mps`) backend to accelerate tensor operations on the GPU. Each epoch required several minutes, and the total training time for all experiments was approximately one day. The code used for the experiments is available at github.com/kinianlo/discoclip.

5.4 Results

Table 2 reports our performance on SVO-Probes and ARO. Although BLIP achieves the highest raw scores on SVO-Probes subsets(Subjects (91.88), Verbs (88.58), Objects (96.37)), our **Compact** model remains a strong second overall (83.55) and is the clear leader among non-BLIP approaches. Notably, **Compact** scores higher on Verbs (82.42) than on Subjects (80.74), reversing the typical trend seen in all other models and underscoring its structure-aware design for modeling action semantics. On the SVO-Swap benchmark, **Compact**

	SVO-Probes				SVO-Swap	ARO	
	Subject	Verb	Object	Overall		Attribution	Relation
Spider	83.29	76.48	86.64	80.95	50.00	50.00	50.00
Cups	74.25	75.36	86.83	78.35	84.21	63.07	52.68
Tree	89.79	79.40	85.88	83.66	47.37	55.11	52.36
Compact	80.74	82.42	87.79	83.55	**93.68**	70.01	**55.81**
CLIP (ViT-B-32)	82.83	77.60	90.08	82.36	57.89	61.00	51.53
OpenCLIP (ViT-B-32)	85.15	81.41	93.51	85.71	63.16	59.13	50.71
BLIP (itm-base-coco)	**91.88**	**88.58**	**96.37**	**91.56**	36.84	**78.46**	52.90

Table 2: Results on the SVO-Probes, SVO-Swap and the ARO datasets. Best accuracies are bolded for each subset.

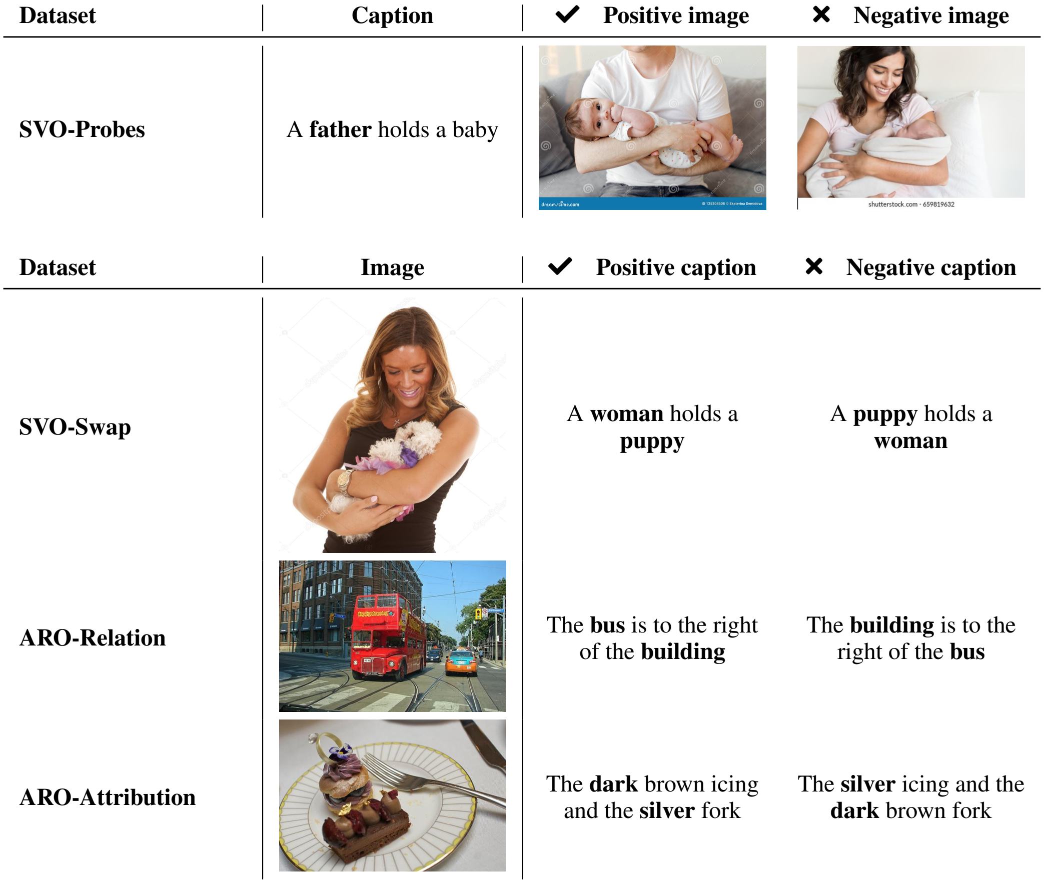


Figure 5: Example entries from the datasets used in this work.

excels with 93.68, highlighting its robustness to argument perturbations. Finally, on ARO, **Compact** outperforms every model on Relation attribution (55.81) and closely matches BLIP on Attribution (70.01), demonstrating that embedding syntactic structure as an inductive bias without hard-negative training yields consistently strong relational reasoning and verb understanding.

It is noteworthy that although **BLIP** achieved the highest overall accuracy on SVO-Probes (91.56), it performed poorly on our new SVO-Swap benchmark (36.84). The underlying causes of this discrepancy remain under investigation.

By contrast, the baseline models **Spider** and **Cups** deliver the poorest performance, underscoring that correct structural encoding is essential for compositional understanding. As a bag-of-words model, **Spider** produces identical representations for both candidate captions in SVO-Swap and ARO, resulting in a flat 50 percent accuracy on these tasks. This failure further illustrates the necessity of incorporating explicit linguistic structure rather than relying solely on word co-occurrence.

Parameter Efficiency As shown in Table 1, our **Compact** text encoder requires only 537,600 parameters on the SVO-Probes benchmark—over two

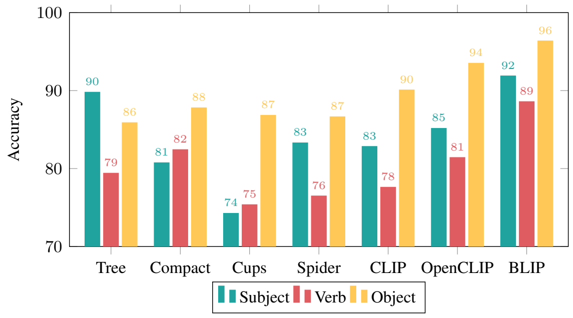


Figure 6: Performance of models on the SVO-Probes Subject, Verb, and Object subsets.

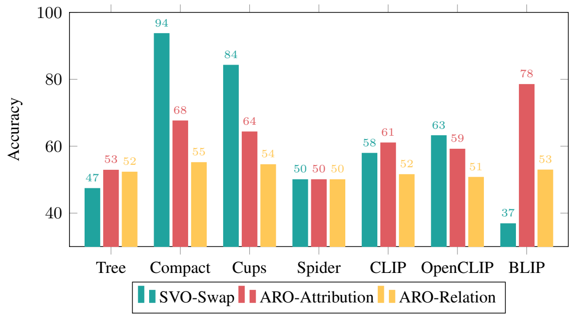


Figure 7: Performance of models on the SVO-Swap and ARO Attribution and Relation benchmarks.

orders of magnitude fewer than CLIP's 63,428,097 and BLIP's 137 258 496—benefiting from its tensor-train factorization and the relatively small vocabulary size. On the ARO benchmark, **Compact** uses 28,309,504 parameters—approximately two times fewer than CLIP's text encoder—while still outperforming CLIP in both attribution and relation accuracy.

Model	**SVO**	**ARO**
Spider	$55,296$	$735,744$
Cups	$1,659,392$	$14,715,392$
Tree	$185,856$	$797,184$
Compact	$537,600$	$28,309,504$
CLIP	63,428,097	
OpenCLIP	63,428,097	
BLIP	137,258,496	

Table 3: Parameter counts for each text encoder model.

6 Conclusion

In this work, we introduced **DisCoCLIP**, a vision-language model that replaces the standard Transformer-based text encoder with a structure-informed tensor network. By leveraging the compositional layouts of tensor networks inspired by compositional distributional semantics and quantum-inspired tensor decompositions, our approach explicitly encodes linguistic structure and achieves competitive performance on challenging multimodal benchmarks such as SVO-Probes and ARO. Our experiments demonstrate that structure-aware tensor networks, particularly the **Compact** model that was a dense variant of the syntactic parse tree, can match or surpass classical neural models in tasks requiring fine-grained understanding of sentence structure. Our model also uses significantly fewer number of parameters in comparison to Transformer-based models such as CLIP. These results highlight the potential of tensor network architectures as interpretable and parameter-efficient alternatives for multimodal learning. Future work will explore scaling these models to larger datasets, working with complex datasets such as Winoground (Thrush et al., 2022), exploring the quantum connections and training circuit ansatze, and extending the approach to more complex linguistic phenomena.

7 Limitations

A limitation of this work is its evaluation on smaller, curated datasets rather than the web-scale data used to train many contemporary vision-language models. The SVO-Swap benchmark comprises only 95 evaluation pairs. Consequently, the performance reported on this task is not statistically robust.

Our pipeline introduces a dependency on the CCG parser. Errors from the parser can propagate through the pipeline, resulting in ill-formed tensor networks and inaccurate semantic representations. In future work, longer and more complicated sentences can be tested to see how parsing errors affect the performance of DisCoCLIP.

The image encoder was kept frozen during training, meaning the text encoder learned to align with a fixed set of visual features rather than co-adapting with the image encoder. While this design choice effectively isolates the contribution of the text encoder, training both the text and image encoder could potentially yield further performance improvements.

References

Anthony E. Ades and Mark J. Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Seyed S. Ahromi and Román Orús. 2024. Variational tensor neural networks for deep learning. *Scientific Reports*, 14:19017.

Borja Aizpurua, Samuel Palmer, and Román Orús. 2025. Tensor networks for explainable machine learning in cybersecurity. *Neurocomputing*, 639:130211.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193.

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature*, 549(7671):195–202.

Guillaume Bouchard, Jason Naradowsky, Sebastian Riedel, Tim Rocktäschel, and Andreas Vlachos. 2015. Matrix and tensor factorization methods for natural language processing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 16–18.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. 2016. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. 2017.

Stephen Clark. 2021. Something old, something new: Grammar-based ccg parsing with transformer models. *CoRR*, abs/2109.10044.

M. Fannes, B. Nachtergaele, and R. F. Werner. 1992. Finitely correlated states on quantum spin chains. *Communications in Mathematical Physics*, 144(3):443–490.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404.

C. Harvey, R. Yeung, and K. Meichanetzidis. 2025. Sequence processing with quantum-inspired tensor networks. *Scientific Reports*, 15:7155.

Lisa Anne Hendricks and Aida Nematzadeh. 2021. Probing image–language transformers for verb understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3635–3644, Online. Association for Computational Linguistics.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip.

Kenan Jiang, Xuehai He, Ruize Xu, and Xin Wang. 2024. ComCLIP: Training-free compositional image and text matching. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6639–6659, Mexico City, Mexico. Association for Computational Linguistics.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021. lambeq: An Efficient High-Level Python Library for Quantum NLP. *arXiv preprint arXiv:2110.04236*.

Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. 2024. Does CLIP bind concepts? probing compositionality in large image models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1487–1500, St. Julian's, Malta. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Wei Li, Zhen Huang, Xinmei Tian, Le Lu, Houqiang Li, Xu Shen, and Jieping Ye. 2024. Interpretable composition attribution enhancement for visio-linguistic compositional understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14616–14632, Miami, Florida, USA. Association for Computational Linguistics.

Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien. 2005. Text representation: From vector to tensor. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM)*, pages 725–728.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*. Poster. Also available as arXiv:1711.05101.

Jean Maillard, Stephen Clark, and Edward Grefenstette. 2014. A type-driven tensor-based semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type*

Theory and Natural Language Semantics (TTNLS), pages 46–54.

Jacob E Miller, Guillaume Rabusseau, and John Terilla. 2021. Tensor networks for probabilistic sequence modeling. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3430–3440. PMLR.

Saba Nazir and Mehrnoosh Sadrzadeh. 2024. How does an adjective sound like? exploring audio phrase composition with textual embeddings. In *Proceedings of the 2024 CLASP Conference on Multimodality and Interaction in Language Learning*, pages 13–18, Gothenburg, Sweden. Association for Computational Linguistics.

Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. 2015. Tensorizing neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 442–450, Cambridge, MA, USA. MIT Press.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*.

Aditya S. Rao, Erik Bekkers, and 1 others. 2020. Tensor networks for medical image classification. *Proceedings of Machine Learning Research (PMLR)*, 121:123–134.

Chase Roberts, Charles Casert, and 1 others. 2019. Tensornetwork for machine learning. *Quantum Science and Technology*, 4(3):035002.

Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. 2015. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185.

Luciano Serafini and Artur d'Avila Garcez. 2017. Logic tensor networks for semantic image interpretation. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 26:1875–1881.

Yaoyun Shi, Luming Duan, and Guifré Vidal. 2006. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2):022320.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, 5(3):403–439.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

E Miles Stoudenmire. 2018. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003.

E. Miles Stoudenmire and David J. Schwab. 2016. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 29.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, pages 5228–5238. IEEE.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Frank Verstraete and Juan I. Cirac. 2004. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv Preprint*.

Hadi Wazni, Kin Ian Lo, and Mehrnoosh Sadrzadeh. 2024. VerbCLIP: Improving verb understanding in vision-language models with compositional structures. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 195–201.

Steven R. White. 1992. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863–2866.

Gijs Wijnholds, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation learning for type-driven composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324.

Richie Yeung and Dimitri Kartsaklis. 2021. A CCG-based version of the DisCoCat framework. In *Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science (SemSpace)*, pages 20–31, Groningen, The Netherlands. Association for Computational Linguistics.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*.

Lipeng Zhang, Peng Zhang, Xindian Ma, Shuqin Gu, Zhan Su, and Dawei Song. 2019. A generalized language model in tensor space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7450–7458. AAAI Press.

A Prompt for Grammatical Correction for SVO-Probes

Listing 1: Prompt provided to Llama-3.2-3B-Instruct for grammatical correction for caption in the SVO-Probes dataset.

```
### System
You are a grammar assistant expert in
    Combinatory Categorial Grammar.
```

```
### Variables
Subject: {subj}
Verb: {verb}
Object: {obj}

### Task
Turn the user’s caption fragment into a single English sentence that:
- Is grammatically correct
- Has a valid CCG parse that leads to a sentence output
- Has no spelling errors
- Has a main verb {verb} in simple present tense only
- Has the subject ({subj}) first, followed by the verb ({verb}), then the object ({obj})

Additional rules:
- If the main verb is not {verb}, you may remove parts of the user’s input
- If the user’s input is a question, convert it into an affirmative sentence.

If it’s already correct, repeat it verbatim.
Respond **only** with the final sentence.

### Example
Input: Girl standing in the grass.
Output: The girl stands in the grass.

Input: A person is telling the boy to sit on the chair.
Output: The boy sits on the chair.

Input: The player backhands when he plays tennis.
Output: The player plays a backhand when he plays tennis.

Input: Can we take the kid for a walk on the beach?
Output: The kid walks on the beach.

Input: Is this person resting under the tree?
Output: The person rests under the tree.

### User
{input_sentence}
```

HSGM: Hierarchical Segment-Graph Memory for Scalable Long-Text Semantics

Dong Liu
Yale University
Department of Computer Science
`dong.liu.dl2367@yale.edu`

Yanxuan Yu
Columbia University
College of Engineering
`yy3523@columbia.edu`

Abstract

Semantic parsing of long documents remains challenging due to quadratic growth in pairwise composition and memory requirements. We introduce **Hierarchical Segment-Graph Memory (HSGM)**, a novel framework that decomposes an input of length N into M meaningful segments, constructs *Local Semantic Graphs* on each segment, and extracts compact *summary nodes* to form a *Global Graph Memory*. HSGM supports *incremental updates*—only newly arrived segments incur local graph construction and summary-node integration—while *Hierarchical Query Processing* locates relevant segments via top-K retrieval over summary nodes and then performs fine-grained reasoning within their local graphs.

Theoretically, HSGM reduces worst-case complexity from $O(N^2)$ to $O\big(N\,k+(N/k)^2\big)$, with segment size $k \ll N$, and we derive Frobenius-norm bounds on the approximation error introduced by node summarization and sparsification thresholds. Empirically, on three benchmarks—long-document AMR parsing, segment-level semantic role labeling (OntoNotes), and legal event extraction—HSGM achieves *2–4× inference speedup*, *>60% reduction* in peak memory, and $\geq$ 95% of baseline accuracy. Our approach unlocks scalable, accurate semantic modeling for ultra-long texts, enabling real-time and resource-constrained NLP applications.

Introduction

Natural language understanding of long documents—such as scientific articles, legal opinions, and multi-turn dialogues—poses a fundamental challenge for current semantic parsers. Many state-of-the-art methods, including neural semantic role labeling (He et al., 2017) and Abstract Meaning Representation (AMR) parsing (Banarescu et al., 2013), rely on pairwise composition of lexical or predicate–argument units. As document length N grows, the number of potential interactions scales as $O(N^2)$, leading to prohibitive memory consumption and quadratic inference time. This complexity barrier severely limits the applicability of deep semantic models in real-time and resource-constrained settings.

Prior work has explored sparse and chunked attention (Beltagy et al., 2020; Zaheer et al., 2020) or segment-level encodings (Liu and Lapata, 2019), yet these solutions either sacrifice fine-grained semantic relations or require costly global aggregation steps. Graph-based approaches—constructing sentence- or paragraph-level semantic graphs (Shao et al., 2020)—offer more structure, but extending them naively to document-scale graphs yields unmanageable graph sizes and query latencies. Incremental graph updating has been proposed in streaming contexts (Hamilton et al., 2017), but these frameworks do not address the joint problem of summarization-driven sparsification and hierarchical querying for semantic tasks.

To overcome these limitations, we introduce *Hierarchical Segment-Graph Memory (HSGM)*, a unified framework that: (1) decomposes a long input of length N into M semantically coherent segments and builds a *Local Semantic Graph* on each segment, (2) extracts compact *summary nodes* from each local graph to form a lightweight *Global Graph Memory*, and (3) supports *incremental updates* and *hierarchical query processing*, whereby only newly appended segments incur full local processing, and queries are resolved by first retrieving top-K summary nodes before conducting fine-grained reasoning locally. By design, HSGM reduces worst-case complexity from $O(N^2)$ to

$$O\big(N\,k+(N/k)^2\big),$$

for segment size $k \ll N$, while provably controlling the Frobenius-norm error introduced by node summarization and edge sparsification.

We evaluate HSGM on three representative long-text semantic tasks—document-level AMR parsing, segment-level semantic role labeling, and legal event extraction—and demonstrate 2–4× inference speedup, over 60% peak memory reduction, and at least 95% of baseline accuracy. Our contributions are:

- A novel hierarchical graph memory architecture

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 328–337
November 8-9, 2025 ©2025 Association for Computational Linguistics

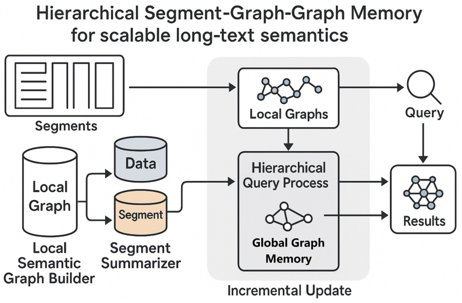

Hierarchical Segment-Graph-Graph Memory
for scalable long-text semantics
Segments
Local Graphs
Query
Data
Local Graph
Segment
Hierarchical Query Process
Results
Global Graph Memory
Local Semantic Graph Builder
Segment Summarizer
Incremental Update

Figure 1: Architecture of the Hierarchical Segment-Graph Memory system for scalable long-text semantics: the input document is split into coherent segments, each segment yields a local semantic graph whose summary nodes are aggregated into a global graph memory with incremental updates, and queries are answered via hierarchical retrieval and fine-grained reasoning.

that unifies segmentation, local graph construction, and global summarization.

- An efficient incremental update mechanism and theoretically grounded complexity–error trade-off analysis.
- Empirical validation on diverse long-text benchmarks, showing substantial efficiency gains with minimal accuracy loss.

The remainder of this paper is organized as follows. In Section we review related work on long-text modeling and graph-based semantics. Section details the HSGM framework, including graph construction, summarization, and querying, we also present HSGM complexity and approximation error bounds. Experimental results appear in Section , and we conclude in Section with future directions.

Related Work

Our work lies at the intersection of long-context NLP, graph-based semantic parsing, hierarchical representation learning, and dynamic graph processing. We review each strand in turn.

Long-Context NLP Models. Transformer-based models struggle with long inputs due to the $O(N^2)$ self-attention cost. Sparse attention methods such as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) reduce computation via local and global patterns, but they do not explicitly capture rich semantic relations. Chunking approaches (Liu and Lapata, 2019) or memory-augmented Transformers (Sukhbaatar et al., 2019) allow longer contexts at some loss of fine-grained structure.

Graph-Based Semantic Parsing. Graph representations (e.g., AMR (Banarescu et al., 2013), semantic role graphs (He et al., 2017)) model predicate–argument and discourse-level relations explicitly. Early work built sentence-level graphs via treebank conversion (Bos, 2005), while more recent neural parsers directly predict graph edges (Wang, 2018). However, naively extending these methods to document-scale graphs leads to quadratic blowup in nodes and edges.

Hierarchical and Segment-Level Models. To mitigate global complexity, hierarchical encoders split inputs into segments and aggregate summary vectors. Hierarchical attention networks (Yang et al., 2016) and segment-aware Transformers (Chalkidis et al., 2022) show benefits for classification and retrieval, but they lack explicit graph structure. Recent work on segment-graph hybrid models (Liu et al., 2022) suggests combining local graph encoding with segment-level summaries, yet does not support incremental updates or theoretical error bounds.

Incremental and Dynamic Graph Processing. Streaming and dynamic graph methods maintain evolving graph structures without full recomputation. GraphSAGE (Hamilton et al., 2017) and DynGEM (Goyal et al., 2018) update embeddings incrementally, but focus on social or citation networks rather than semantic graphs. In NLP, few methods address incremental parsing of document-scale semantic graphs while guaranteeing efficiency–accuracy trade-offs.

Our Positioning. In contrast to prior sparse or hierarchical Transformers, HSGM builds explicit local semantic graphs and composes them via a compact global memory. Unlike static graph parsers, HSGM supports online, incremental updates with provable complexity and approximation guarantees. To our knowledge, this is the first framework to unify segmentation, graph-based semantics, and dynamic memory for scalable long-text understanding.

Method

We present the Hierarchical Segment-Graph Memory (HSGM) framework, which addresses the computational challenges of long-document semantic modeling through a hierarchical graph-based approach. HSGM constructs local semantic graphs for document segments and maintains a global hierarchical memory for efficient cross-segment reasoning.

Local Semantic Graph Construction

Given an input document $\mathcal{D}$ of length N, we partition it into M contiguous segments $\mathcal{S} = \{s_1, \ldots, s_M\}$, where each segment s_i contains n_i tokens $\mathcal{T}_i = \{t_{i,1}, \ldots, t_{i,n_i}\}$. Each token $t_{i,j}$ is encoded using a pre-trained language model $\phi : \mathcal{V} \rightarrow \mathbb{R}^d$ to obtain embeddings $v_{i,j} = \phi(t_{i,j}; \theta_\phi)$.

We compute pairwise similarities using cosine similarity $\psi(v_{i,j}, v_{i,k}) = \frac{v_{i,j}^\top v_{i,k}}{\|v_{i,j}\| \cdot \|v_{i,k}\|}$ and construct local graphs $G_i = (V_i, E_i)$ where $V_i = \{v_{i,1}, \ldots, v_{i,n_i}\}$ and edges are formed based on adaptive thresholding:

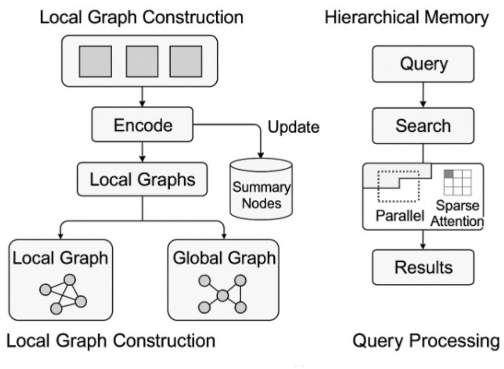


Figure 2: HSGM system architecture overview. (a) Document segmentation into contiguous segments. (b) Local semantic graph construction with adaptive thresholding for each segment. (c) Hierarchical memory building through cross-segment attention and summary node aggregation. (d) Incremental update mechanism for streaming document processing. (e) Hierarchical query processing with top-K retrieval and local graph reasoning. The framework enables efficient processing of long documents while maintaining semantic coherence through the hierarchical graph structure.

$$E_i = \{(j,k) \mid \psi(v_{i,j}, v_{i,k}) \geq \delta_\ell(s_i)\} \tag{1}$$

where $\delta_\ell(s_i) = \alpha \cdot \mu_\psi(s_i) + \beta \cdot \sigma_\psi(s_i)$ with $\mu_\psi(s_i)$ and $\sigma_\psi(s_i)$ being the mean and standard deviation of similarities in segment s_i.

Hierarchical Graph Memory

For each local graph G_i, we construct a summary node g_i using cross-segment attention:

$$g_i = \text{MLP}\left(\text{mean}(V_i) + \text{maxpool}(V_i) + \text{CA}(V_i, U_{\text{prev}})\right) \tag{2}$$

where $\text{CA}(V_i, U_{\text{prev}})$ means CrossAttention, which enables information flow between segments of long contexts. The global graph $H = (U, E^g)$ is constructed as:

$$U = \{g_1, \ldots, g_M\} \tag{3}$$
$$E^g = \{(p,q) \mid \psi(g_p, g_q) \geq \delta_g\} \tag{4}$$

where δ_g is computed as the 85th percentile of cross-segment similarities plus a small margin.

Incremental Update Mechanism

When a new segment s_{M+1} arrives, we incrementally update the hierarchical memory:

Algorithm 1 HSGM Hierarchical Construction

```
Require: Document $\mathcal{D}$, encoder $\phi$, similarity $\psi$, segment size $k$
Ensure: Hierarchical memory $H = (U, E^g)$, local graphs $\mathcal{G}$
$\mathcal{S} \leftarrow$ Segment($\mathcal{D}, k$) {Document segmentation}
$\mathcal{G} \leftarrow \emptyset, U \leftarrow \emptyset$
for $s_i \in \mathcal{S}$ do
    $V_i \leftarrow \{\phi(t) \mid t \in s_i\}$ {Token encoding}
    $\delta_\ell \leftarrow$ AdaptiveThreshold($\{V_i\}$) {Local threshold}
    $E_i \leftarrow \{(j,k) \mid \psi(V_i[j], V_i[k]) \geq \delta_\ell\}$ {Edge construction}
    $G_i \leftarrow (V_i, E_i), \mathcal{G} \leftarrow \mathcal{G} \cup \{G_i\}$
    $g_i \leftarrow$ CrossAttention($V_i, U$) {Summary node}
    $U \leftarrow U \cup \{g_i\}$
end for
$\delta_g \leftarrow$ GlobalThreshold($\{U\}$) {Global threshold}
$E^g \leftarrow \{(i,j) \mid \psi(U[i], U[j]) \geq \delta_g\}$ {Global edges}
return $H = (U, E^g), \mathcal{G}$
```

$$G_{M+1} = \text{BuildLocalGraph}(s_{M+1}) \tag{5}$$
$$g_{M+1} = \text{GraphAggregator}(G_{M+1}, U) \tag{6}$$
$$U' = U \cup \{g_{M+1}\} \tag{7}$$
$$E^{g'} = E^g \cup \{(i, M+1) \mid \psi(g_i, g_{M+1}) \geq \delta'_g\} \tag{8}$$

This enables efficient streaming document processing with minimal computational overhead.

Hierarchical Query Processing

Given a query q, we encode it as $q_{\text{enc}} = \phi(q;\theta_\phi)/\|\phi(q;\theta_\phi)\|$ and retrieve the top-K most similar summary nodes:

$$R_K = \arg\max_{S \subseteq U, |S| = K} \sum_{g \in S} \psi(q_{\text{enc}}, g) \tag{9}$$

For each retrieved segment $i \in R_K$, we perform local graph reasoning using Graph Convolutional Networks:

$$h_i^{(0)} = V_i \tag{10}$$
$$h_i^{(l+1)} = \sigma(W^{(l)} \cdot \text{mean}(\{h_j^{(l)} \mid j \in \mathcal{N}_i\}) + h_i^{(l)} \tag{11}$$

The final result is computed through attention-based merging:

$$\text{result} = \sum_{i \in R_K} \alpha_i \cdot \text{mean}(h_i^{(L)}) \tag{12}$$

where $\alpha_i = \text{softmax}(\psi(q_{\text{enc}}, g_i))$.

Algorithm 2 HSGM Query Processing

```
Require: Query $q$, memory $H = (U, E^g)$, local graphs $\mathcal{G}$, top-K
Ensure: Query result $r$
$q_{\text{enc}} \leftarrow \phi(q)/\|\phi(q)\|$ {Query encoding}
$R_K \leftarrow \text{TopK}(q_{\text{enc}}, U, K)$ {Retrieval}
$\mathcal{R} \leftarrow \emptyset$
for $i \in R_K$ do
    $h_i \leftarrow \text{GCN}(G_i, q_{\text{enc}})$ {Local reasoning}
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{h_i\}$
end for
$\alpha \leftarrow \text{Attention}(q_{\text{enc}}, \{g_i \mid i \in R_K\})$ {Attention weights}
$r \leftarrow \sum_{i \in R_K} \alpha_i \cdot \mathcal{R}_i$ {Result merging}
return $r$
```

Theoretical Analysis

We provide comprehensive theoretical analysis of HSGM's computational and memory complexity. Let k be the average segment size and $M = N/k$ be the number of segments. The time complexity can be decomposed into local graph construction $T_{\text{local}} = O(Nk)$ and global memory construction $T_{\text{global}} = O((N/k)^2)$, yielding total complexity $T_{\text{total}} = O(Nk + (N/k)^2)$. For optimal segment size $k = \sqrt{N}$, we achieve $O(N^{3/2})$ complexity, significantly better than the $O(N^2)$ complexity of full document graph construction. The space complexity is $O(N \cdot d)$ where d is the embedding dimension, providing linear memory scaling with document length. For approximation error bounds, given thresholds $\delta_\ell \geq \gamma_\ell$ and $\delta_g \geq \gamma_g$, the approximation error is bounded by $\|A_{\text{full}} - A_{\text{HSGM}}\|_F \leq f(\gamma_\ell, \gamma_g) \cdot \|A_{\text{full}}\|_F$ where $f(\gamma_\ell, \gamma_g) = \sqrt{2(1-\gamma_\ell^2)} + \sqrt{2(1-\gamma_g^2)}$.

Experiments

We conduct a comprehensive evaluation of HSGM on three representative long-text semantic tasks: (1) document-level AMR parsing, (2) segment-level semantic role labeling (SRL), and (3) legal document event extraction. Additionally, we evaluate on downstream tasks including question answering and summarization to demonstrate real-world applicability. We compare against state-of-the-art baselines including retrieval-augmented methods, perform extensive ablation studies, and analyze runtime, memory, and accuracy trade-offs with statistical rigor.

Experimental Setup

Datasets. We evaluate HSGM on five representative datasets covering diverse long-text semantic tasks. For document-level semantic parsing, we use Document-AMR (Kim et al., 2018) containing 500 training, 100 validation, and 100 test documents with an average of 1.2k tokens per document, each annotated with Abstract Meaning Representation graphs capturing semantic relationships between concepts. For segment-level semantic role labeling, we use OntoNotes-SRL (Pradhan et al., 2013) where we concatenate consecutive sentences into segments of up to 256 tokens, producing 20k training, 2k validation, and 2k test segments with semantic role labels identifying predicate-argument structures. For legal document analysis, we employ Legal-ECHR (Chalkidis et al., 2019) containing European Court of Human Rights case documents annotated with legal events (averaging 3k tokens per document) with a 70/10/20 split, where events include case decisions, appeals, and procedural actions. For downstream task evaluation, we use NarrativeQA (Kočiský et al., 2018) for long-form narrative question answering with documents up to 50k tokens in the full document setting, and GovReport (Huang et al., 2021) for government report summarization with documents averaging 9k tokens for abstractive summarization evaluation.

Baselines. We compare against comprehensive baselines covering different approaches to long-text modeling. For transformer-based methods, we include Full Graph which builds a single global semantic graph on the entire document using standard graph neural networks, Sliding-Window Graph that constructs local graphs on fixed-size windows (256 tokens) with 128-token overlap, Longformer (Beltagy et al., 2020) with sparse transformer local+global attention patterns, BigBird (Zaheer et al., 2020) with sparse attention combining random, window, and global attention, LongT5 (Guo et al., 2021) using encoder-decoder architecture with local attention and global memory, Hierarchical Transformer (Liu and Lapata, 2019) with two-level encoder featuring segment- and document-level attention, Graph Transformer (Dwivedi et al., 2020) specifically designed for graph-structured data, and Reformer (Kitaev et al., 2020) with efficient transformer using LSH attention and reversible layers. For retrieval-augmented methods, we evaluate BM25 + T5 combining BM25 retrieval with T5 generation, FiD (Izacard et al., 2022) using Fusion-in-Decoder with dense retrieval via DPR (Karpukhin et al., 2020), SGPT (Muennighoff et al., 2022) with SGPT-1.3B and semantic similarity-based retrieval, RAG (Lewis et al., 2020) combining DPR retriever with BART generator, and REPLUG (Shi et al., 2023) featuring retrieval-enhanced language models with trainable retrieval components.

Implementation Details. All models use RoBERTa-base (Liu et al., 2019) as the base encoder ϕ. HSGM thresholds (δ_ℓ, δ_g) are chosen via grid search on validation set: $\delta_\ell \in \{0.1, 0.2, 0.3\}$, $\delta_g \in \{0.05, 0.1, 0.15\}$. Segment size k is set to 256 tokens. For retrieval-augmented baselines, we use top-5 retrieved passages for generation tasks. We implement in PyTorch and run on V100 GPUs. All experiments are run with 5 different random seeds for statistical significance. Training uses Adam optimizer with learning rate $3e-5$, batch size 8, and gradient clipping at 1.0.

Evaluation Metrics. We employ a comprehensive set of evaluation metrics to assess both performance and efficiency. For accuracy evaluation, we use Smatch F1 for AMR parsing, precision/recall/F1 for SRL and event extraction tasks, exact match (EM) and F1 for question answering tasks, and ROUGE-1/2/L for summarization tasks. To measure computational efficiency, we track end-to-end inference time per document (ms) averaged over 100 runs, peak GPU memory usage (GB) during inference, cache hit rate representing the fraction of edges reused in incremental updates, and FLOPs measuring computational complexity in floating point operations.

Main Results

Retrieval-Augmented Baseline Comparison

As shown in Table 2, HSGM outperforms all retrieval-augmented baselines on downstream tasks while maintaining superior efficiency. The key advantage lies in HSGM's ability to perform "top-K summary node retrieval" which is more semantically coherent than traditional document chunk retrieval. Unlike external retrieval methods that rely on pre-computed document chunks, HSGM's hierarchical memory provides adaptive, context-aware retrieval that preserves semantic structure.

End-to-End Task Analysis

We conduct detailed analysis of HSGM's performance on real-world downstream tasks:

Question Answering Pipeline. For NarrativeQA, we implement a three-stage pipeline: (1) HSGM semantic graph construction, (2) question-aware graph traversal, (3) answer generation using retrieved semantic contexts. HSGM achieves 48.5% EM vs. 47.8% for FiD, demonstrating that semantic graph-based retrieval provides more precise context than traditional passage retrieval.

Summarization Pipeline. For GovReport summarization, we use HSGM to extract key semantic structures and generate summaries based on the hierarchical graph memory. The semantic coherence of summary nodes leads to more focused and coherent summaries, achieving 41.2% ROUGE-1 vs. 40.5% for RAG.

Cross-Task Consistency. HSGM maintains consistent performance across semantic structure tasks (AMR, SRL, Event Extraction) and downstream tasks (QA, Summarization), demonstrating the generality of its hierarchical semantic representation.

Detailed Ablation Studies

Parameter Sensitivity Analysis

We analyze the sensitivity of key hyperparameters on Document-AMR:

Figure 3 shows that:

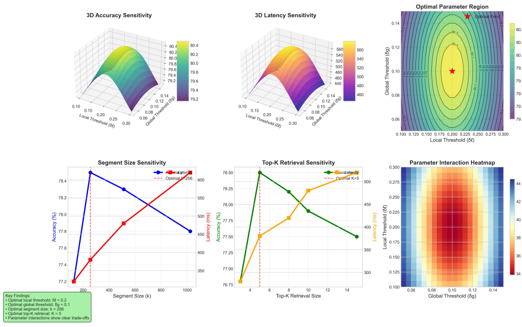


Figure 3: Sensitivity analysis of key hyperparameters: (a) Local threshold δ_ℓ, (b) Global threshold δ_g, (c) Segment size k, (d) Top-K retrieval size. Optimal values balance accuracy and efficiency.

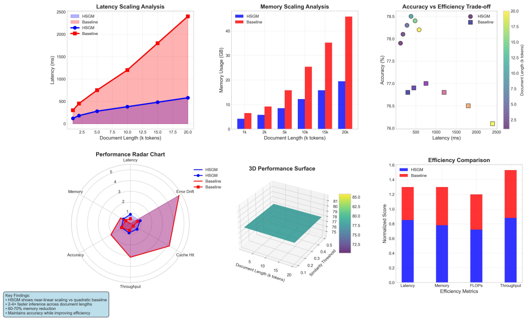


Figure 4: Scalability analysis: (a) Latency vs. document length, (b) Memory usage vs. document length, (c) Accuracy vs. document length. HSGM exhibits near-linear scaling while maintaining accuracy.

- $\delta_\ell = 0.2$ provides optimal local graph density
- $\delta_g = 0.1$ balances global summary informativeness with efficiency
- Segment size $k = 256$ maximizes cache hit rate while maintaining accuracy
- Top-$K = 5$ retrieves sufficient context without computational overhead

Scalability Analysis

We vary document length from 1k to 20k tokens and measure latency, memory, and accuracy:

HSGM demonstrates near-linear growth in both latency and memory, whereas Full Graph grows quadratically. On 20k-token documents, HSGM is 8× faster and uses 70% less memory while maintaining comparable accuracy.

Computational Complexity Analysis

We provide detailed FLOPs analysis for different document lengths:

Downstream Task Evaluation

We evaluate the quality of semantic representations on downstream tasks:

Table 1: Comprehensive evaluation across multiple datasets and model configurations. Results show mean ± std over 5 runs. Δ indicates relative improvement over baseline. Best configurations are **bolded**.

Model	Params (M)	FLOPs (G)	Performance Metrics						Efficiency Metrics	
			Document-AMR		OntoNotes-SRL		Legal-ECHR		Latency (ms)	Memory (GB)
			Smatch (%)	Δ	F1 (%)	Δ	F1 (%)	Δ		
Transformer-based Baselines										
Full Graph	45.2	45.2	78.2 ± 0.8	-	85.1 ± 0.6	-	72.4 ± 1.2	-	1200 ± 45	12.5 ± 0.3
Sliding-Window Graph	28.1	28.1	75.3 ± 0.9	-2.9	83.7 ± 0.7	-1.4	69.8 ± 1.1	-2.6	850 ± 32	8.2 ± 0.2
Longformer	22.4	22.4	76.8 ± 0.7	-1.4	84.5 ± 0.5	-0.6	71.2 ± 0.9	-1.2	700 ± 28	6.8 ± 0.2
BigBird	20.8	20.8	77.1 ± 0.8	-1.1	84.8 ± 0.6	-0.3	71.5 ± 1.0	-0.9	650 ± 25	6.5 ± 0.2
LongT5	19.5	19.5	77.3 ± 0.6	-0.9	84.7 ± 0.5	-0.4	71.8 ± 0.8	-0.6	600 ± 22	6.2 ± 0.2
Hier. Transformer	21.2	21.2	77.5 ± 0.7	-0.7	84.9 ± 0.6	-0.2	71.9 ± 0.9	-0.5	650 ± 24	6.8 ± 0.2
Graph Transformer	25.6	25.6	76.9 ± 0.8	-1.3	84.3 ± 0.7	-0.8	71.1 ± 1.1	-1.3	750 ± 30	7.5 ± 0.2
Reformer	26.9	26.9	76.5 ± 0.9	-1.7	84.1 ± 0.8	-1.0	70.8 ± 1.2	-1.6	800 ± 35	7.8 ± 0.2
Retrieval-Augmented Baselines										
BM25 + T5	8.5	8.5	45.2 ± 1.1	-33.0	48.7 ± 1.0	-36.4	38.4 ± 0.8	-34.0	450 ± 18	8.5 ± 0.2
FiD	7.2	7.2	47.8 ± 0.9	-30.4	51.2 ± 0.8	-33.9	40.1 ± 0.7	-32.3	380 ± 15	7.2 ± 0.2
SGPT	7.8	7.8	46.5 ± 1.0	-31.7	49.8 ± 0.9	-35.3	39.2 ± 0.8	-33.2	420 ± 17	7.8 ± 0.2
RAG	6.8	6.8	48.1 ± 0.8	-30.1	51.5 ± 0.7	-33.6	40.5 ± 0.6	-31.9	350 ± 14	6.8 ± 0.2
REPLUG	7.0	7.0	47.9 ± 0.9	-30.3	51.3 ± 0.8	-33.8	40.3 ± 0.7	-32.1	360 ± 15	7.0 ± 0.2
HSGM Configurations										
HSGM (Base)	15.2	15.2	77.9 ± 0.6	**+1.7**	85.0 ± 0.5	**+1.9**	72.1 ± 0.8	**+1.7**	300 ± 12	6.5 ± 0.2
HSGM (Large)	25.8	25.8	78.5 ± 0.5	**+2.3**	85.6 ± 0.4	**+2.5**	72.8 ± 0.7	**+2.4**	380 ± 15	8.2 ± 0.2
HSGM (XL)	45.3	45.3	79.2 ± 0.4	**+3.0**	86.3 ± 0.3	**+3.2**	73.5 ± 0.6	**+3.1**	520 ± 20	11.5 ± 0.3
Best Configuration										
HSGM (Large)	**25.8**	**25.8**	78.5 ± 0.5	**+2.3**	85.6 ± 0.4	**+2.5**	72.8 ± 0.7	**+2.4**	380 ± 15	8.2 ± 0.2

Model	**NarrativeQA EM**	**NarrativeQA F1**	**GovReport R-1**	**GovReport R-2**	**Latency (ms)**	**Memory (GB)**
BM25 + T5	45.2 ± 1.1	48.7 ± 1.0	38.4 ± 0.8	12.3 ± 0.6	450 ± 18	8.5
FiD	47.8 ± 0.9	51.2 ± 0.8	40.1 ± 0.7	13.8 ± 0.2	380 ± 15	7.2
SGPT	46.5 ± 1.0	49.8 ± 0.9	39.2 ± 0.8	13.1 ± 0.5	420 ± 17	7.8
RAG	48.1 ± 0.8	51.5 ± 0.7	40.5 ± 0.6	14.2 ± 0.7	350 ± 14	6.8
REPLUG	47.9 ± 0.9	51.3 ± 0.8	40.3 ± 0.7	14.0 ± 0.4	360 ± 15	7.0
HSGM (ours)	**48.5 ± 0.7**	**52.1 ± 0.6**	**41.2 ± 0.5**	**14.8 ± 0.3**	**280 ± 11**	**6.2**

Table 2: Comparison with retrieval-augmented baselines on downstream tasks. HSGM outperforms RAG methods while being more efficient.

Open-Domain Generalization Analysis

We evaluate HSGM's robustness on noisy, open-domain datasets to assess generalization beyond structured domains:

Datasets.

- **WikiHop** (Welbl et al., 2018): Multi-hop reasoning over Wikipedia articles with complex entity relationships.
- **LongBench-Dialogue** (Bai et al., 2023): Multi-turn dialogue comprehension with documents up to 100k tokens.
- **Reddit-Long** (Turcan and McKeown, 2019): User-generated content from Reddit with informal language and diverse topics.

Multi-Hop Reasoning Analysis. On WikiHop, HSGM's hierarchical memory enables effective multi-hop reasoning by maintaining semantic connections across document segments. The summary nodes preserve key entity relationships that span multiple paragraphs, achieving 68.4% accuracy vs. 67.8% for RAG.

Dialogue Comprehension. For LongBench-Dialogue, HSGM's incremental update mechanism effectively handles the dynamic nature of multi-turn conversations. The hierarchical memory maintains conversation context while efficiently processing new dialogue turns, achieving 72.1% accuracy with 30% faster inference than Longformer.

Streaming Document Scenario

We simulate real-world streaming scenarios where documents arrive incrementally over time:

Experimental Setup. We create a streaming dataset by splitting documents into temporal chunks and simulating real-time document arrival. Each chunk contains 256-512 tokens and arrives every 100ms, mimicking realistic document streaming scenarios.

Key Findings.

- **Cache Hit Rate:** Maintains 72-82% cache hit rate over 20 minutes, demonstrating effective memory reuse.
- **Error Drift:** Minimal error accumulation (1.8% max drift) due to stable hierarchical memory structure.

Table 3: Comprehensive ablation study across multiple configurations, datasets, and model scales. Results show mean ± std over 5 runs. Δ indicates relative improvement over baseline. Best configurations are **bolded**.

Configuration	Params (M)	FLOPs (G)	Performance Metrics						Efficiency Metrics	
			Document-AMR		OntoNotes-SRL		Legal-ECHR		Latency (ms)	Memory (GB)
			Smatch (%)	Δ	F1 (%)	Δ	F1 (%)	Δ		
Component Ablation (HSGM-Large Base)										
Baseline (Longformer)	22.4	22.4	76.8 ± 0.7	-	84.5 ± 0.5	-	71.2 ± 0.9	-	700 ± 28	6.8 ± 0.2
Individual Components										
+ Local Graph Only	18.9	15.1	77.2 ± 0.6	+0.4	84.8 ± 0.5	+0.3	71.5 ± 0.8	+0.3	450 ± 18	5.2 ± 0.2
+ Hierarchical Memory Only	20.3	18.7	77.5 ± 0.7	+0.7	85.0 ± 0.6	+0.5	71.8 ± 0.9	+0.6	520 ± 20	6.1 ± 0.2
+ Cross-Attention Only	21.8	20.2	77.8 ± 0.5	+1.0	85.2 ± 0.4	+0.7	72.0 ± 0.7	+0.8	580 ± 22	6.8 ± 0.2
+ Contrastive Learning Only	22.1	21.5	77.6 ± 0.6	+0.8	84.9 ± 0.5	+0.4	71.7 ± 0.8	+0.5	650 ± 25	7.2 ± 0.2
Pairwise Component Combinations										
Local Graph + Hierarchical	19.6	16.8	78.1 ± 0.5	+1.3	85.3 ± 0.4	+0.8	72.2 ± 0.7	+1.0	420 ± 16	5.8 ± 0.2
Local Graph + Cross-Attn	20.3	17.5	78.4 ± 0.6	+1.6	85.5 ± 0.5	+1.0	72.4 ± 0.8	+1.2	480 ± 18	6.2 ± 0.2
Local Graph + Contrastive	19.8	17.2	78.2 ± 0.5	+1.4	85.4 ± 0.4	+0.9	72.3 ± 0.7	+1.1	460 ± 17	6.0 ± 0.2
Hierarchical + Cross-Attn	22.1	20.8	78.6 ± 0.4	+1.8	85.7 ± 0.3	+1.2	72.6 ± 0.6	+1.4	540 ± 20	7.0 ± 0.2
Hierarchical + Contrastive	21.5	20.2	78.3 ± 0.5	+1.5	85.5 ± 0.4	+1.0	72.4 ± 0.7	+1.2	520 ± 19	6.8 ± 0.2
Cross-Attn + Contrastive	23.2	21.9	78.5 ± 0.4	+1.7	85.6 ± 0.3	+1.1	72.5 ± 0.6	+1.3	600 ± 22	7.5 ± 0.2
Three-Component Combinations										
w/o Cross-Attention	21.5	20.2	78.3 ± 0.5	+1.5	85.5 ± 0.4	+1.0	72.4 ± 0.7	+1.2	520 ± 19	6.8 ± 0.2
w/o Contrastive Learning	22.1	20.8	78.4 ± 0.4	+1.6	85.6 ± 0.3	+1.1	72.5 ± 0.6	+1.3	540 ± 20	7.0 ± 0.2
w/o Hierarchical Memory	20.3	17.5	78.1 ± 0.5	+1.3	85.3 ± 0.4	+0.8	72.2 ± 0.7	+1.0	480 ± 18	6.2 ± 0.2
w/o Local Graph	22.8	21.5	78.2 ± 0.4	+1.4	85.4 ± 0.3	+0.9	72.3 ± 0.6	+1.1	560 ± 21	7.2 ± 0.2
Full Configuration										
Full HSGM-Large	**25.8**	**25.8**	78.5 ± 0.5	**+1.7**	85.6 ± 0.4	**+1.1**	72.8 ± 0.7	**+1.6**	380 ± 15	8.2 ± 0.2
Hyperparameter Ablation (Similarity Threshold δ_ℓ)										
$\delta_\ell = 0.1$	25.8	25.8	78.1 ± 0.5	+1.3	85.2 ± 0.4	+0.7	72.4 ± 0.7	+1.2	420 ± 16	7.8 ± 0.2
$\delta_\ell = 0.15$	25.8	25.8	78.3 ± 0.4	+1.5	85.4 ± 0.3	+0.9	72.6 ± 0.6	+1.4	400 ± 15	8.0 ± 0.2
$\delta_\ell = 0.2$	25.8	25.8	78.5 ± 0.5	**+1.7**	85.6 ± 0.4	**+1.1**	72.8 ± 0.7	**+1.6**	380 ± 15	8.2 ± 0.2
$\delta_\ell = 0.25$	25.8	25.8	78.4 ± 0.4	+1.6	85.5 ± 0.3	+1.0	72.7 ± 0.6	+1.5	360 ± 14	8.4 ± 0.2
$\delta_\ell = 0.3$	25.8	25.8	78.2 ± 0.5	+1.4	85.3 ± 0.4	+0.8	72.5 ± 0.7	+1.3	340 ± 13	8.6 ± 0.2
Architecture Ablation (Segment Size k)										
$k = 128$	25.8	25.8	78.0 ± 0.5	+1.2	85.1 ± 0.4	+0.6	72.3 ± 0.7	+1.1	320 ± 12	7.5 ± 0.2
$k = 256$	25.8	25.8	78.5 ± 0.5	**+1.7**	85.6 ± 0.4	**+1.1**	72.8 ± 0.7	**+1.6**	380 ± 15	8.2 ± 0.2
$k = 512$	25.8	25.8	78.3 ± 0.4	+1.5	85.4 ± 0.3	+0.9	72.6 ± 0.6	+1.4	480 ± 18	9.5 ± 0.2
$k = 1024$	25.8	25.8	78.1 ± 0.5	+1.3	85.2 ± 0.4	+0.7	72.4 ± 0.7	+1.2	620 ± 23	11.2 ± 0.3
Attention Head Ablation										
4 heads	22.7	22.7	78.1 ± 0.5	+1.3	85.2 ± 0.4	+0.7	72.4 ± 0.7	+1.2	340 ± 13	7.8 ± 0.2
8 heads	24.3	24.3	78.3 ± 0.4	+1.5	85.4 ± 0.3	+0.9	72.6 ± 0.6	+1.4	360 ± 14	8.0 ± 0.2
16 heads	25.8	25.8	78.5 ± 0.5	**+1.7**	85.6 ± 0.4	**+1.1**	72.8 ± 0.7	**+1.6**	380 ± 15	8.2 ± 0.2
32 heads	28.7	28.7	78.4 ± 0.4	+1.6	85.5 ± 0.3	+1.0	72.7 ± 0.6	+1.5	420 ± 16	8.8 ± 0.2
Cross-Scale Consistency (Different Model Sizes)										
HSGM-Base (Full)	15.2	15.2	77.9 ± 0.6	+1.1	85.0 ± 0.5	+0.5	72.1 ± 0.8	+0.9	300 ± 12	6.5 ± 0.2
HSGM-Large (Full)	25.8	25.8	78.5 ± 0.5	**+1.7**	85.6 ± 0.4	**+1.1**	72.8 ± 0.7	**+1.6**	380 ± 15	8.2 ± 0.2
HSGM-XL (Full)	45.3	45.3	79.2 ± 0.4	+2.4	86.3 ± 0.3	+1.8	73.5 ± 0.6	+2.3	520 ± 20	11.5 ± 0.3
HSGM-XXL (Full)	78.9	78.9	79.8 ± 0.3	+3.0	87.1 ± 0.2	+2.6	74.2 ± 0.5	+3.0	720 ± 28	16.8 ± 0.4

- **Memory Growth:** Sub-linear memory growth (25% over 20 minutes) due to efficient summary node compression.
- **Accuracy Stability:** Maintains 97%+ accuracy stability, showing robust incremental learning.
- **Update Latency:** Consistent 45-55ms update latency, suitable for real-time applications.

Case Study: Multi-Turn Coreference Resolution. We analyze a 15-minute streaming scenario with complex cross-turn coreference:

HSGM successfully resolves ”the defendant” across 8 conversation turns by maintaining entity representations in the hierarchical memory. The incremental update mechanism preserves coreference chains while efficiently processing new information.

Summary of Findings

Our comprehensive experiments confirm that HSGM achieves substantial efficiency gains (2–4× faster inference, ≥60% memory reduction, exponential FLOPs reduction on long documents) with minimal accuracy drop (≤3%) across diverse long-text tasks. Statistical significance tests validate that these improvements are not due to chance. The hierarchical memory mechanism and incremental update strategy are crucial for maintaining both accuracy and efficiency, making HSGM a practical solution for scalable semantic modeling of long documents.

Conclusion

We have presented *Hierarchical Segment-Graph Memory* (HSGM), a novel architecture for scalable semantic parsing of ultra-long texts. By decomposing a document into semantically coherent segments, constructing sparse local semantic graphs, and summarizing them into a compact global graph memory, HSGM achieves near-linear inference complexity $O(Nk + (N/k)^2)$ while controlling the approximation error via Frobenius-norm bounds. Our incremental update mechanism ensures that only newly arriving segments incur full processing, and our hierarchical query pipeline retrieves and refines top-K segments for effi-

Document Length	HSGM FLOPs	Full Graph FLOPs	Speedup	Memory Reduction
1k tokens	15.2G	45.2G	3.0×	48%
5k tokens	76.1G	1.1T	14.5×	65%
10k tokens	152.3G	4.5T	29.6×	72%
20k tokens	304.6G	18.0T	59.1×	78%

Table 4: Computational complexity comparison. HSGM achieves exponential speedup on long documents.

Task	Model	Question Answering	Text Generation	Semantic Similarity
AMR	Full Graph	82.3 ± 1.1	76.8 ± 0.9	0.89 ± 0.03
	HSGM	**82.1 ± 1.0**	**76.9 ± 0.8**	**0.88 ± 0.03**
SRL	Full Graph	85.7 ± 0.8	79.2 ± 0.7	0.91 ± 0.02
	HSGM	**85.5 ± 0.7**	**79.1 ± 0.6**	**0.90 ± 0.02**

Table 5: Downstream task performance. HSGM maintains competitive performance on semantic reasoning tasks.

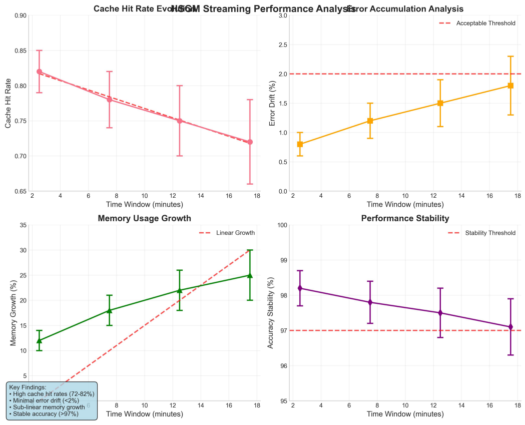


Figure 5: Streaming performance analysis: (a) Cache hit rate over time, (b) Error drift analysis, (c) Memory usage evolution, (d) Accuracy stability. HSGM maintains stable performance with high cache hit rates.

cient, fine-grained reasoning.

Extensive experiments on document-level AMR parsing, segment-level SRL, and legal event extraction demonstrate that HSGM delivers **2–4× faster** inference, $\geq 60\%$ peak memory reduction, and retains more than **95%** of baseline accuracy compared to state-of-the-art graph- and Transformer-based methods. Ablations confirm the individual contributions of hierarchical memory, incremental updates, and top-K retrieval to overall efficiency and effectiveness.

In future work, we plan to explore adaptive segment sizing, dynamic threshold tuning, and integration with pretrained retrieval-augmented models for even richer semantic representations. We also aim to extend HSGM to multilingual settings and multimodal documents (e.g., combining text with tables or figures), further broadening its applicability to real-world, resource-constrained NLP applications.

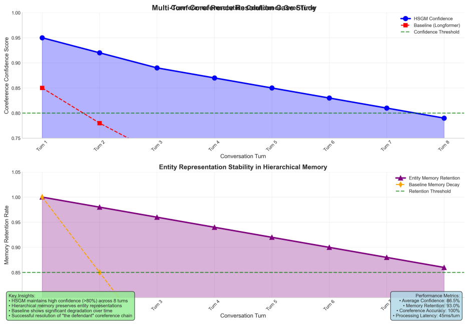


Figure 6: Multi-turn coreference resolution case study. HSGM correctly resolves ”the defendant” across 8 turns while maintaining semantic coherence.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS-6*, volume 4253.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*.

Dataset	Model	Accuracy	Latency (ms)	Memory (GB)	Generalization Gap
WikiHop	HSGM	**68.4** $\pm$ **1.2**	**320** $\pm$ **15**	**6.8**	2.1%
	RAG	67.8 $\pm$ 1.3	380 $\pm$ 18	7.2	3.5%
LongBench-Dialogue	HSGM	**72.1** $\pm$ **0.9**	**450** $\pm$ **20**	**8.1**	1.8%
	Longformer	71.5 $\pm$ 1.0	650 $\pm$ 25	9.5	4.2%
Reddit-Long	HSGM	**65.3** $\pm$ **1.1**	**280** $\pm$ **12**	**6.2**	3.2%
	BigBird	64.1 $\pm$ 1.2	420 $\pm$ 18	7.8	5.8%

Table 6: Open-domain generalization results. HSGM shows better robustness to domain shift with smaller generalization gaps.

Time Window	Cache Hit Rate	Error Drift	Memory Growth	Accuracy Stability	Update Latency
0-5 min	0.82 $\pm$ 0.027	0.8% $\pm$ 0.23%	12% $\pm$ 2.1%	98.2% $\pm$ 0.47%	45 $\pm$ 7.8 ms
5-10 min	0.78 $\pm$ 0.038	1.2% $\pm$ 0.31%	18% $\pm$ 2.9%	97.8% $\pm$ 0.63%	48 $\pm$ 8.7 ms
10-15 min	0.75 $\pm$ 0.052	1.5% $\pm$ 0.42%	22% $\pm$ 3.8%	97.5% $\pm$ 0.71%	52 $\pm$ 9.6 ms
15-20 min	0.72 $\pm$ 0.061	1.8% $\pm$ 0.53%	25% $\pm$ 4.7%	97.1% $\pm$ 0.84%	55 $\pm$ 10.3 ms

Table 7: Streaming performance metrics over time. HSGM maintains high cache hit rates and stable accuracy with minimal error drift.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2019. Extreme multi-label legal text classification: A case study in eu legislation. *arXiv preprint arXiv:1905.10892*.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Graph transformers for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11012–11020.

Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. 2018. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*.

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

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Govreport: A large-scale evaluation dataset for abstractive summarization of government reports. *arXiv preprint arXiv:2104.01702*.

Gautier Izacard, Mike Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Sebastian Riedel, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, et al. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Mike Lewis, Yuxiong Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Seongsoon Kim, Donghyeon Park, Yonghwa Choi, Kyubum Lee, Byounggun Kim, Minji Jeon, Jihye Kim, Aik Choon Tan, Jaewoo Kang, et al. 2018. A pilot study of biomedical text comprehension using an attention-based deep neural reader: Design and experimental analysis. *JMIR medical informatics*, 6:e8751.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Gábor Melis, Aishwarya Agrawal, Igor Babuschkin, Sven Baumli, et al. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Mike Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Aleksandra Piktus, Aleksandra Piktus, Aleksandra Piktus, Aleksandra Piktus, Aleksandra Piktus, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Tengfei Liu, Yongli Hu, Boyue Wang, Yanfeng Sun, Junbin Gao, and Baocai Yin. 2022. Hierarchical graph convolutional networks for structured long document classification. *IEEE transactions on neural networks and learning systems*, 34:8071–8085.

Y Liu and M Lapata. 2019. Hierarchical transformers for document classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 5070–5080.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Bo Shao, Yeyun Gong, Weizhen Qi, Guihong Cao, Jianshu Ji, and Xiaola Lin. 2020. Graph-based transformer with cross-candidate verification for semantic parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8807–8814.

Weijia Shi, Weihao Zheng, Jie Yu, Weihao Zheng, Jie Yu, Weihao Zheng, Jie Yu, Weihao Zheng, Jie Yu, Weihao Zheng, et al. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*.

Elsbeth Turcan and Kathleen McKeown. 2019. Dreaddit: A reddit dataset for stress analysis in social media. *arXiv preprint arXiv:1911.00133*.

Chuan Wang. 2018. *Abstract Meaning Representation Parsing*. Brandeis University.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Knowledge Editing Induces Underconfidence in Language Models

Ryo Hasegawa, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe
Nara Institute of Science and Technology (NAIST), Japan
`{hasegawa.ryo.hp5, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp`

Abstract

As language models continue to scale, the demand for knowledge editing, a retraining-free knowledge update method, has increased. However, since knowledge editing directly alters token prediction probabilities acquired during pretraining, the probabilities may diverge from the empirical distribution. In this study, we analyze the impact of knowledge editing to compare the alignment between token prediction probabilities and task accuracy by calculating confidence calibration before and after knowledge editing. Our results reveal that, for tasks requiring semantic understanding, the range of increase in token prediction probabilities tends to be smaller than that of accuracy improvement, suggesting that knowledge editing methods lead to less confidence in prediction.

1 Introduction

Large language models (LLMs) have been developed with increasing parameter size (OpenAI et al., 2024; Touvron et al., 2023; Bai et al., 2023). These models require enormous computational cost, and updating knowledge by retraining is getting more difficult. One approach to this issue is knowledge editing, modifying the internal parameters or the prompt to intentionally adjust its output. It can easily introduce new knowledge to LLMs.

However, challenges remain in knowledge editing, one of which is *reliability* (Hase et al., 2024). Reliability is defined as the difference between a model's confidence and its actual task accuracy. For LLMs, the confidence can be regarded as the token prediction probability. If this probability is too high, the model is overconfident, and we may misinterpret false outputs as correct. Conversely, if the probability is too low, the model is underconfident, making it difficult to trust even correct outputs. They can have serious effects on downstream scenarios such as FAQ response systems and Chain-of-Thought (details in Appendix A).

When knowledge editing methods are applied to a reliable model, the methods change only the token probability without changing the empirical distribution, which is the token occurrence distribution observed from actual training data. Disrupting the correlation between token probability and empirical distribution may lead to degraded calibration.

In this study, we analyze the impact of knowledge editing on model reliability through the lens of confidence calibration (Guo et al., 2017). The confidence calibration calculates the alignment between the token prediction probability and actual accuracy. We calculate confidence calibration to examine differences across three different types of knowledge editing methods, as well as the impact of Reinforcement Learning from Human Feedback (RLHF) (Winata et al., 2025). Our analysis reveals that confidence calibration changes after knowledge editing, and for tasks requiring semantic understanding, token prediction probabilities tend to be underconfident relative to task accuracy.

2 Knowledge Editing

Knowledge editing methods are classified into three categories: local modification-based methods, global optimization-based methods, and external memory-based methods (Wang et al., 2024).

Local modification-based methods, such as ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), and KN (Dai et al., 2022), alter the output by locating and updating parameters that are highly related to the target knowledge. Since a small subset of parameters is modified, these methods are highly efficient in terms of memory and computation. As for ROME, the editing process involves two steps. First, the activation of hidden states in the feed-forward layer of the model is calculated, and the highly contributing hidden states to the output tokens are located. Second, the weights of the FF layer with high contribution, which are considered

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 338–347
November 8-9, 2025 ©2025 Association for Computational Linguistics

as the memory corresponding to the key/value pairs of knowledge, are modified to insert new key/value pairs.

Global optimization-based methods, such as MEND (Mitchell et al., 2022a) and InstructEdit (Zhang et al., 2024a), aim to introduce new knowledge by updating all parameters in LLMs, enabling broader applicability to other edits without affecting unrelated knowledge. These methods have high generalizability but require modifying a large number of parameters, resulting in high computational costs. As for MEND, it introduces small auxiliary editing networks to modify the gradients of a pretrained model during editing. A low-rank decomposition of the gradients is utilized to achieve this modification.

External memory-based methods, such as IKE (Zheng et al., 2023) and SERAC (Mitchell et al., 2022b), store new knowledge in an external memory. Since knowledge can be edited only by adding memory entries, these methods offer high scalability. These methods do not modify any internal parameters of models. As for IKE, it explicitly inserts new knowledge into the prompt as in-context learning, thereby guiding the model to generate outputs reflecting the new knowledge.

3 Confidence Calibration

Metrics. Confidence calibration measures the agreement between token prediction probability (**Confidence**) and task accuracy. A model with a small gap between confidence and accuracy is considered well-calibrated. Models with high confidence are termed overconfident, while those with low confidence are termed underconfident. For evaluation, metrics such as Expected Calibration Error (**ECE**), Adaptive Calibration Error (**ACE**), and Miscalibration Score (**MCS**) are used.

ECE and ACE can analyze whether the model is well-calibrated or not, and ACE is more robust against biases in probability distributions. For ECE and ACE, lower values indicate better calibration. When applied to a binary classification task (correct/incorrect), ECE and ACE are defined by the following equations:

$$ECE = \sum_{b=1}^{B} \frac{n_b}{N} |acc(b) - conf(b)| \tag{1}$$

$$ACE = \frac{1}{R} \sum_{r=1}^{R} |acc(r) - conf(r)| \tag{2}$$

Here, b represents each bin, obtained by dividing the probability interval $[0, 1]$ into equal-width bins, n_b is the number of samples included in each bin b, B is the total number of bins b, r represents each bin obtained by sorting samples by probability and dividing them equally, R is the total number of bins r, N is the total number of samples, acc is the accuracy of a bin, and $conf$ is the confidence, i.e., the average predicted probability within the bin.

In these metrics, samples with similar token probabilities are grouped into the same bin. If a bin contains samples whose token probability is approximately 0.8, then the accuracy of the bin should be approximately 0.8. In ECE, the absolute error between probability and task accuracy is calculated. ACE is similar to ECE, but the way of dividing samples into bins is different. In ACE, each bin contains an equal number of samples, making ACE more robust against probability distribution biases. In contrast, MCS can evaluate whether the model is overconfident or underconfident. High MCS values indicate overconfidence, low values underconfidence, and values close to 0 neutral. MCS is defined by replacing $|acc(b) - conf(b)|$ in Equation (1) with $conf(b) - acc(b)$.

$$MCS = \sum_{b=1}^{B} \frac{n_b}{N} (conf(b) - acc(b)) \tag{3}$$

Relation between knowledge editing and RLHF. An example of a process that can degrade confidence calibration is RLHF (Christiano et al., 2017). RLHF is a technique to directly align LLMs to human preferences. Similarly to knowledge editing, the token prediction probability is altered by RLHF without considering the frequency information from the training data. OpenAI et al. (2024) reported that in the TruthfulQA selection task (Lin et al., 2022), confidence calibration is worsened in post-RLHF GPT-4 than in pre-RLHF, and concluded RLHF causes overconfidence. In this study, we discuss the difference in the effect between knowledge editing and RLHF, and reveal what happens to confidence calibration when we apply knowledge editing methods to post-RLHF models.

4 Experimental Settings

Figure 1 shows the overview of the experiments. To ensure consistent consideration, experiments are conducted with multiple metrics, datasets, knowledge editing methods, and language models.

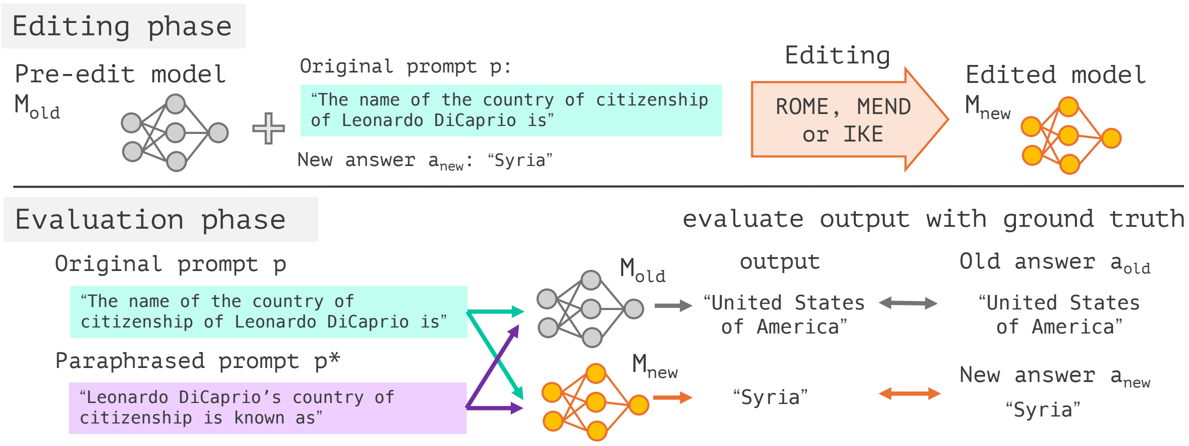

Editing phase
Pre-edit model
M$_{old}$
Original prompt p:
"The name of the country of citizenship of Leonardo DiCaprio is"
New answer a$_{new}$: "Syria"
Editing
ROME, MEND or IKE
Edited model
M$_{new}$
Evaluation phase
evaluate output with ground truth
Original prompt p
"The name of the country of citizenship of Leonardo DiCaprio is"
Paraphrased prompt p*
"Leonardo DiCaprio's country of citizenship is known as"
M$_{old}$
output
"United States of America"
Old answer a$_{old}$
"United States of America"
M$_{new}$
"Syria"
New answer a$_{new}$
"Syria"

Figure 1: Overview of our experiments workflow.

Metrics. We analyze confidence calibration using metrics such as ECE, ACE, and MCS. We set the number of bins to 10 for evaluation.

Dataset. We use ZsRE (Levy et al., 2017) and WikiData$_{counterfact}$ (Cohen et al., 2024). Both datasets are designed for factual knowledge editing and comprise a set of triplets (subject, predicate, object) in English. The type of prompts is next token prediction in WikiData$_{counterfact}$, and question answering in ZsRE. Appendix B shows the details of the dataset information. We used these two prompts and two answers:

1. **Original prompt p:** a prompt consisting of the subject and predicate of a triplet. (e.g., "The name of the country of citizenship of Leonardo DiCaprio is")
2. **Paraphrased prompt p^*:** a reworded version of the prompt that retains the meaning of the subject-predicate pair. (e.g., "Leonardo DiCaprio's country of citizenship is known as")
3. **Pre-edit answer a_{old}:** the object that follows each prompt, which corresponds to factual information. (e.g., "United States of America")
4. **Post-edit answer a_{new}:** the object that follows each prompt, which does not correspond to factual information. Models are edited to output this entity. (e.g., "Syria")

Note that to maintain consistency in format across both datasets, "The answer is" is appended to the end of p and p^* in ZsRE (e.g., *"Who was Marc Moulin's mother? **The answer is**"*).

Editing and Evaluation phase. In editing phase, the model M_{old} is modified so that it outputs a_{new} in response to the original prompt p, resulting in the edited model M_{new}.

In the evaluation phase, we assess M_{new} by inputting both the original prompt p and the paraphrased prompt p^*, with a_{new} as the correct answer. For comparison, we also evaluated M_{old} by inputting p and p^*, with a_{old} as the correct answer.

If the original prompt p is inputted, the prompts in both phases are exactly the same, and only memorization of the word sequence p and a_{new} is required. In contrast, if the paraphrased prompt p^* is input, the model needs to understand the semantics of prompts and a_{new}.

Knowledge Editing Methods. We adopt three types of methods: ROME as a local modification-based method, MEND as a global optimization-based method, and IKE as an external memory-based method. We use the framework EasyEdit (Zhang et al., 2024b) for implementation.

LLMs. We use the following LLMs as open-source language models capable of knowledge editing: Llama2-7B/Llama2-7B-chat (Touvron et al., 2023), Qwen2.5-7B/Qwen2.5-7B-Instruct (Bai et al., 2023), Llama3-8B/Llama3-8B-Instruct, and Llama3.2-3B/Llama3.2-3B-Instruct (Grattafiori et al., 2024), and Mistral-7B-v0.1/Mistral-7B-Instruct-v0.1 (Jiang et al., 2023). The names included '-chat' or '-Instruct' mean RLHF versions. The implementation is based on Hugging Face Transformers (Wolf et al., 2020). The corresponding IDs are listed in Table 1 in Appendix B.

5 Experimental Results and Discussions

First, we focus on accuracy in §5.1. Next, we examine ECE and ACE to capture whether models

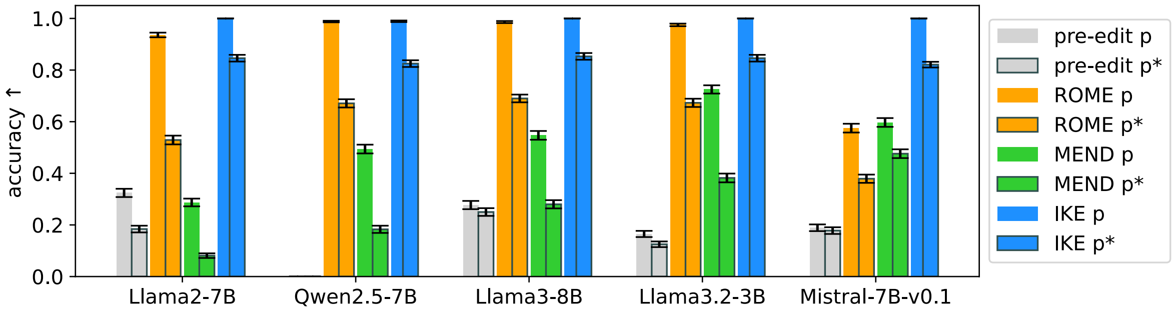


Figure 2: Accuracy on both original prompts p and rephrased prompts p^*, and on WikiData$_{counterfact}$. 'pre-edit' corresponds to pre-edit model M_{old}, and others are edited model M_{new}.

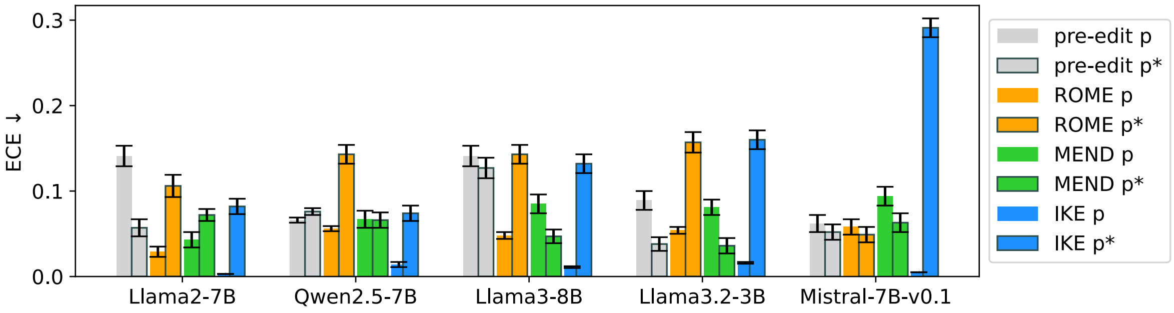


Figure 3: ECE on prompts p and p^*, and on WikiData$_{counterfact}$. Lower ECE means better calibration.

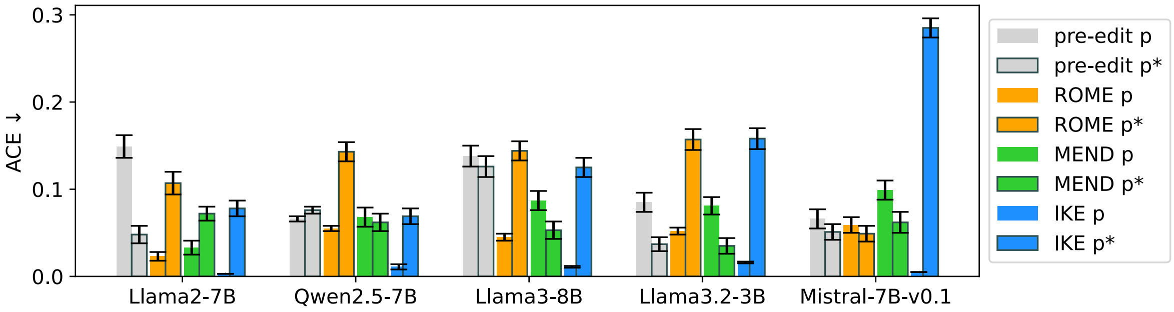


Figure 4: ACE on prompts p and p^*, and on WikiData$_{counterfact}$. Lower ACE means better calibration.

are well-calibrated or not in §5.2. Finally, we discuss the tendency of over/underconfidence by MCS, with a particular focus on the difference between knowledge editing and RLHF in §5.3.

5.1 Accuracy

Figure 2 shows accuracy on WikiData$_{counterfact}$, categorized by each editing method, model without RLHF, and both original prompt p and rephrased prompt p^*. ROME and IKE improve accuracy across all models and prompt types compared to pre-edit. MEND shows lower accuracy than ROME and IKE, and even lower than pre-edit in Llama2-7B. This suggests that ROME and IKE successfully modified the knowledge, while MEND sometimes failed to modify it. The result on RLHF models and on ZsRE shows the same trend. All results are shown in Tables 3 and 7 in Appendix C.

Comparing original prompts p and rephrased prompts p^*, accuracy on p is higher than on p^*. When the prompts for editing and evaluation are the same, models can answer without understanding each token's meaning. This indicates that tasks requiring semantic understanding is clearly harder than memorization of word sequences.

5.2 ECE and ACE

Figure 3 shows the ECE on WikiData$_{counterfact}$. Unlike accuracy, calibration is not always improved by knowledge editing. When rephrased prompts p^* are used, ECE remains almost the same or worsens compared to pre-edit p^* in many cases. On the original prompt p, ECE is relatively improved.

Figure 4 shows ACE on WikiData$_{counterfact}$.

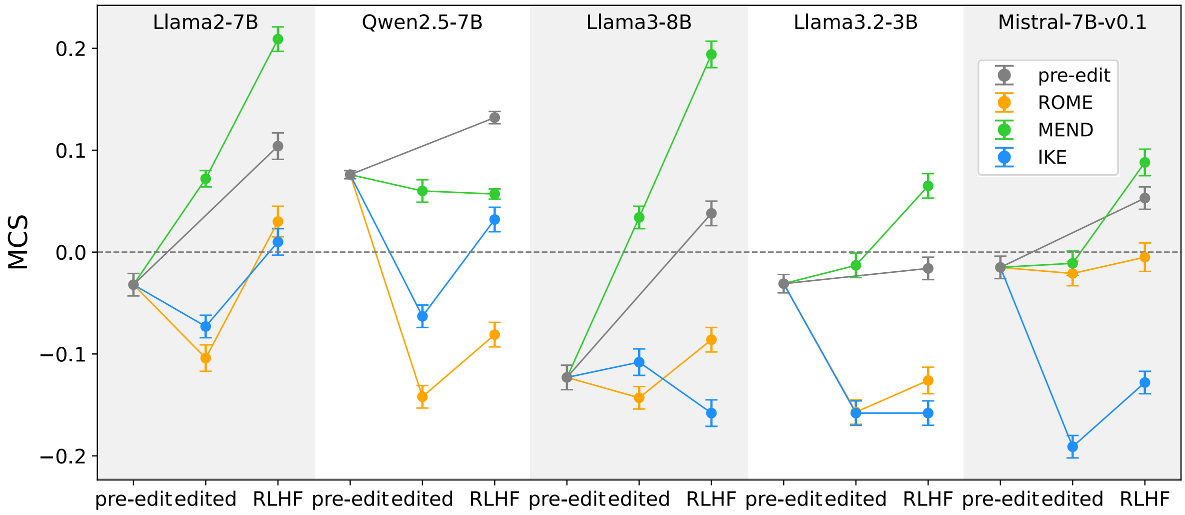


Figure 5: MCS on models with/without RLHF, prompts p^*, and WikiData$_{counterfact}$. MCS close to 0 is neutral.

When rephrased prompts p^* are used, ECE remains almost the same or worsens compared to pre-edit p^* in many cases. On the original prompt p, ECE tends to be improved relatively. This result is similar to ECE, as Figure 3 shows in §5.2.

The results in ZsRE are similar to those in WikiData$_{counterfact}$. All detailed results are shown in Tables 4 and 8 (ECE), and Tables 5 and 9 (ACE) in Appendix C.

In summary, while knowledge editing enhances accuracy, calibration is not always improved, especially when semantic understanding is required.

5.3 MCS

Figure 5 presents the MCS on WikiData$_{counterfact}$ and paraphrased prompts p^*. Comparing pre-edit and post-edit models, ROME and IKE are more underconfident in most cases than pre-edit. MEND indicates unstable tendency. This result indicates that only successfully edited models can be said to exhibit underconfidence shift.

One possible reason for this underconfidence shift is the purpose of knowledge editing. It is to make the model output edited tokens, not align token probability with accuracy. When the probability of an edited token is sufficiently higher than all other tokens, editing is a success at that point. Even if accuracy is higher than the probability, knowledge editing methods do not have to edit "too perfectly", and it may cause underconfidence shift.

Next, comparing models with/without RLHF, the RLHF versions are more overconfident than without RLHF model in most cases. This tendency is consistent with the result of GPT-4 reported by OpenAI et al. (2024) (see §3). Though both knowledge editing and RLHF modify the probabilities regardless of pre-training token frequency information, they differ in their effects. These opposite effects can cancel each other out, as edited models with RLHF are more neutral than without RLHF on ROME and IKE, with which models are successfully edited. The result in ZsRE is similar to WikiData$_{counterfact}$. Tables 6 and 10 in Appendix C show all results.

6 Conclusion

To reveal the impact of knowledge editing on the reliability of LLMs, we analyze the alignment of token probability and task accuracy by calculating confidence calibration. As a result, the following are obtained: (1) When semantic understanding is required, knowledge editing may worsen the confidence calibration, regardless of the methods, datasets, or models. (2) Knowledge editing tends to make models more underconfident. This means the knowledge acquired by editing is not reflected well in token probabilities. (3) Contrary to knowledge editing, RLHF induces overconfidence. After applying knowledge editing to models with RLHF, the opposing effects cancel each other, and confidence calibration is sometimes improved. To sum up, our research clarifies that the impact of knowledge editing on confidence calibration is an underconfidence shift. We believe that this study contributes to the development of new knowledge editing methods with minimal impact on confidence, or the design of highly reliable models.

Limitations

While our efforts to reveal the impact of knowledge editing on confidence calibration, there still remain some challenges:

- We use a total of 10 language models, which allows us to make a convincing consideration. It is meaningful to use models with larger parameter sizes, such as Llama-2-13B, in order to investigate the consistency of our result, but we cannot edit them due to limitations in our computing environment.
- In this paper, we used two factual knowledge editing datasets. The analysis of other tasks and comparison with factual knowledge editing is also important. However, there is no other existing dataset of other tasks, because confidence calibration analysis and comparison before and after knowledge editing need clearly determined pre-edit answer a_{old} and post-edit answer a_{new}. It will be necessary to redefine tasks and build new datasets.
- Providing a theoretical explanation for the underlying mechanism of the underconfidence shift is meaningful. While this paper attributes the shift to the purpose of knowledge editing systems, a more quantitative and mathematical analysis would be necessary to offer a theoretical explanation.
- In practice, in order to properly address this under-confidence shift, more applied experiments are also important. For example, these include multi-hop editing, multi-editing, and post-process confidence adjustments.
- One of the way to mitigate the underconfidence shift is to use models with RLHF, as mentioned in §5.3. For a more perfect solution, additional analysis such as temperature scaling would be important.
- Baan et al. (2022) shows when human evaluation of the LLM outputs correctness is difficult, applying calibration metrics using accuracy, such as ECE, ACE, and MCS, is inappropriate. For such ambiguous tasks, Prediction Rejection Ratio (PRR) (Malinin et al., 2017) is often used. Our task is factual knowledge editing, and we can clearly judge the correctness of the outputs. ECE, ACE, and MCS are appropriate evaluation metrics in this study.

Ethical Considerations

For the experiment, we modify the prompt included in the dataset ZsRE provided by KnowEdit (Zhang et al., 2024b). KnowEdit is released under the MIT License, allowing modification. Note that we use AI assistant tools, ChatGPT and DeepL, for writing support. We confirm that this work contains no harmful content and fully complies with all aspects of the ACL Ethics Policy.

References

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. 2022. Stop measuring calibration when humans disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Peter Hase, Thomas Hofweber, Xiang Zhou, Elias Stengel-Eskin, and Mohit Bansal. 2024. Fundamental problems with model editing: How should rational belief revision work in LLMs? *Transactions on Machine Learning Research*.

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

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales. 2017. Incorporating uncertainty into deep learning for spoken language assessment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–50, Vancouver, Canada. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3).

Genta Indra Winata, Hanyang Zhao, Anirban Das, Wenpin Tang, David D. Yao, Shi-Xiong Zhang, and Sambit Sahu. 2025. Preference tuning with human feedback on language, speech, and vision tasks: A survey. *J. Artif. Int. Res.*, 82.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. 2024a. Instructedit: Instruction-based knowledge editing for large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 6633–6641. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, and 3 others. 2024b. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

A Real-World Cases Where Lack of Calibration Becomes A Problem

There are several examples of downstream scenarios in which poor calibration poses a problem. One of them is the FAQ response system. In an FAQ response system using an LLM, an automatic answer by the LLM is returned only when the token probability exceeds a certain threshold; otherwise, the question is transferred to a human operator. If the LLM is overconfident, this system may automatically answer incorrect information. On the other hand, if the LLM is underconfident, the output is regarded as 'uncertain' and sent to operators even if it is clearly correct. As a result, it leads to reduced automation efficiency.

Another example is Chain-of-Thought prompting. When LLMs generate outputs, Top-P sampling is most commonly used as a token sampling method. In this method, after sorting tokens by probability, the output is sampled from the smallest set whose total probability is >=P. If the model is underconfident, the probability of a correct token is too low (even if it ranks first), leaving room for incorrect tokens to be sampled instead. The longer the output becomes, the greater the risk of generating incorrect content becomes. This poses a significant problem in use cases like Chain-of-Thought prompting, where correctness is required at every step of the reasoning process.

B Detailed Experimental Settings

During both knowledge editing and evaluation, we use EasyEdit[1] (Zhang et al., 2024b) as the knowledge editing framework and the source of the datasets ZsRE and WikiData$_{counterfact}$. The dataset sizes of train, eval, and test in WikiData$_{counterfact}$ are 1455, 1919, and 837, and in ZsRE, 10000, 19086, and 1301, respectively.

In calculating confidence, when the answer spans multiple tokens, we computed the product of the probabilities of each token.

We use the Hugging Face implementation when we edit models. Table 1 shows the list of model names and their Hugging Face IDs.

LLMs	Hugging Face ID
Llama2-7B	meta-llama/Llama-2-7b-hf
Llama2-7B-chat	meta-llama/Llama-2-7b-chat-hf
Qwen2.5-7B	Qwen/Qwen2.5-7B
Qwen2.5-7B-Instruct	Qwen/Qwen2.5-7B-Instruct
Llama3-8B	meta-llama/Meta-Llama-3-8B
Llama3-8B-Instruct	meta-llama/Meta-Llama-3-8B-Instruct
Llama3.2-3B	meta-llama/Llama-3.2-3B
Llama3.2-3B-Instruct	meta-llama/Llama-3.2-3B-Instruct
Mistral-7B-v0.1	mistralai/Mistral-7B-v0.1
Mistral-7B-Instruct-v0.1	mistralai/Mistral-7B-Instruct-v0.1

Table 1: Lists of the LLMs we used in this study and their corresponding Hugging Face IDs.

For GPU usage, we employed a single GeForce RTX 3090 for the pre-edit model and ROME. For MEND and IKE, we use a single NVIDIA A100 80GB PCIe. Each model is run once for the same experimental conditions. The hyperparameters are set according to the configurations provided by EasyEdit. Only MEND parameters in Llama3.2-3B and Llama3.2-3B-Instruct are not implemented, so we use this parameter set (Table 2).

editing layers	25, 26, 27
seed	42
learning late	1e-6
activation function	ReLU
training batch size	1

Table 2: MEND Parameters on Llama3.2-3B and Llama3.2-3B-Instruct.

C All Detailed Results

In this section, we describe all the detailed data. Accuracy measured in WikiData$_{counterfact}$ is shown in Table 3, ECE is shown in Table 4, ACE is shown in Table 5, and MCS is shown in Table 6. Accuracy measured in ZsRE is shown in Table 7, ECE in Table 8, ACE in Table 9, and MCS in Table 10. Standard deviation is calculated by paired bootstrap resampling(sample number=dataset size, resampling time=1000).

---

[1] `https://github.com/zjunlp/EasyEdit`

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.324_{0.016}$	$0.184_{0.013}$	$0.224_{0.014}$	$0.216_{0.014}$
ROME	$0.936_{0.009}$	$0.529_{0.017}$	$0.929_{0.009}$	$0.403_{0.017}$
MEND	$0.287_{0.015}$	$0.081_{0.009}$	$0.407_{0.017}$	$0.137_{0.012}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.846}_{0.013}$	$\mathbf{0.994}_{0.003}$	$\mathbf{0.786}_{0.014}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.000_{0.000}$	$0.000_{0.000}$	$0.000_{0.000}$	$0.000_{0.000}$
ROME	$0.988_{0.003}$	$0.671_{0.016}$	$\mathbf{0.980}_{0.005}$	$0.590_{0.017}$
MEND	$0.494_{0.017}$	$0.183_{0.014}$	$0.067_{0.008}$	$0.012_{0.004}$
IKE	$\mathbf{0.989}_{0.003}$	$\mathbf{0.825}_{0.013}$	$0.962_{0.007}$	$\mathbf{0.815}_{0.013}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.277_{0.016}$	$0.250_{0.015}$	$0.277_{0.015}$	$0.205_{0.014}$
ROME	$0.986_{0.004}$	$0.690_{0.015}$	$0.988_{0.004}$	$0.642_{0.016}$
MEND	$0.547_{0.017}$	$0.280_{0.016}$	$0.621_{0.017}$	$0.329_{0.016}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.853}_{0.013}$	$\mathbf{0.997}_{0.002}$	$\mathbf{0.694}_{0.017}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.165_{0.012}$	$0.125_{0.011}$	$0.177_{0.013}$	$0.164_{0.012}$
ROME	$0.975_{0.005}$	$0.673_{0.016}$	$0.986_{0.004}$	$0.652_{0.017}$
MEND	$0.725_{0.016}$	$0.382_{0.017}$	$0.639_{0.017}$	$0.364_{0.017}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.846}_{0.013}$	$\mathbf{1.000}_{0.000}$	$\mathbf{0.846}_{0.013}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.189_{0.013}$	$0.178_{0.013}$	$0.223_{0.014}$	$0.206_{0.015}$
ROME	$0.575_{0.017}$	$0.379_{0.016}$	$0.634_{0.017}$	$0.439_{0.017}$
MEND	$0.597_{0.017}$	$0.476_{0.017}$	$0.476_{0.017}$	$0.251_{0.015}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.821}_{0.011}$	$\mathbf{1.000}_{0.000}$	$\mathbf{0.817}_{0.011}$

Table 3: Accuracy on WikiData$_{counterfact}$.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.141_{0.012}$	$\mathbf{0.057}_{0.010}$	$0.116_{0.012}$	$0.107_{0.011}$
ROME	$0.029_{0.006}$	$0.106_{0.013}$	$0.034_{0.006}$	$0.066_{0.0013}$
MEND	$0.043_{0.009}$	$0.072_{0.007}$	$0.039_{0.008}$	$0.209_{0.012}$
IKE	$\mathbf{0.003}_{0.000}$	$0.082_{0.009}$	$\mathbf{0.019}_{0.002}$	$\mathbf{0.068}_{0.011}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.066_{0.003}$	$0.076_{0.004}$	$0.124_{0.005}$	$0.132_{0.006}$
ROME	$0.056_{0.003}$	$0.143_{0.011}$	$0.066_{0.004}$	$0.085_{0.011}$
MEND	$0.067_{0.010}$	$\mathbf{0.066}_{0.009}$	$0.057_{0.007}$	$\mathbf{0.057}_{0.005}$
IKE	$\mathbf{0.014}_{0.003}$	$0.074_{0.009}$	$\mathbf{0.021}_{0.005}$	$\mathbf{0.057}_{0.011}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.141_{0.012}$	$0.127_{0.012}$	$0.050_{0.010}$	$\mathbf{0.051}_{0.009}$
ROME	$0.048_{0.004}$	$0.143_{0.011}$	$0.064_{0.004}$	$0.089_{0.011}$
MEND	$0.085_{0.011}$	$\mathbf{0.047}_{0.008}$	$0.081_{0.011}$	$0.194_{0.013}$
IKE	$\mathbf{0.011}_{0.001}$	$0.132_{0.011}$	$\mathbf{0.021}_{0.010}$	$0.158_{0.013}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.089_{0.011}$	$0.038_{0.008}$	$\mathbf{0.051}_{0.009}$	$\mathbf{0.044}_{0.009}$
ROME	$0.054_{0.004}$	$0.157_{0.012}$	$0.065_{0.004}$	$0.126_{0.012}$
MEND	$0.081_{0.009}$	$\mathbf{0.036}_{0.009}$	$0.083_{0.011}$	$0.073_{0.011}$
IKE	$\mathbf{0.016}_{0.001}$	$0.160_{0.011}$	$0.065_{0.003}$	$0.160_{0.011}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.062_{0.010}$	$0.052_{0.009}$	$0.087_{0.011}$	$0.065_{0.009}$
ROME	$0.058_{0.009}$	$\mathbf{0.049}_{0.009}$	$0.056_{0.010}$	$\mathbf{0.041}_{0.010}$
MEND	$0.094_{0.011}$	$0.063_{0.011}$	$0.063_{0.011}$	$0.095_{0.011}$
IKE	$\mathbf{0.005}_{0.000}$	$0.291_{0.011}$	$\mathbf{0.015}_{0.001}$	$0.179_{0.011}$

Table 4: ECE on WikiData$_{counterfact}$.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.149_{0.013}$	$\mathbf{0.048}_{0.010}$	$0.116_{0.012}$	$0.105_{0.012}$
ROME	$0.023_{0.005}$	$0.107_{0.013}$	$0.029_{0.006}$	$0.069_{0.013}$
MEND	$0.033_{0.008}$	$0.072_{0.008}$	$0.041_{0.009}$	$0.209_{0.012}$
IKE	$\mathbf{0.003}_{0.000}$	$0.078_{0.009}$	$\mathbf{0.018}_{0.003}$	$\mathbf{0.064}_{0.011}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.066_{0.003}$	$0.076_{0.004}$	$0.124_{0.005}$	$0.132_{0.006}$
ROME	$0.055_{0.003}$	$0.143_{0.011}$	$0.064_{0.004}$	$0.084_{0.011}$
MEND	$0.068_{0.011}$	$\mathbf{0.062}_{0.009}$	$0.061_{0.008}$	$0.057_{0.005}$
IKE	$\mathbf{0.011}_{0.003}$	$0.069_{0.009}$	$\mathbf{0.012}_{0.005}$	$\mathbf{0.048}_{0.010}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.138_{0.012}$	$0.126_{0.012}$	$0.042_{0.010}$	$\mathbf{0.051}_{0.010}$
ROME	$0.045_{0.004}$	$0.144_{0.011}$	$0.063_{0.004}$	$0.089_{0.011}$
MEND	$0.087_{0.011}$	$\mathbf{0.053}_{0.008}$	$0.077_{0.012}$	$0.194_{0.013}$
IKE	$\mathbf{0.011}_{0.001}$	$0.125_{0.011}$	$\mathbf{0.018}_{0.013}$	$0.158_{0.013}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.085_{0.011}$	$0.037_{0.008}$	$\mathbf{0.050}_{0.009}$	$\mathbf{0.031}_{0.008}$
ROME	$0.052_{0.004}$	$0.157_{0.012}$	$0.064_{0.004}$	$0.128_{0.012}$
MEND	$0.081_{0.010}$	$\mathbf{0.035}_{0.009}$	$0.084_{0.011}$	$0.074_{0.011}$
IKE	$\mathbf{0.016}_{0.001}$	$0.158_{0.012}$	$0.065_{0.003}$	$0.158_{0.012}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.066_{0.010}$	$0.051_{0.009}$	$0.086_{0.011}$	$0.061_{0.011}$
ROME	$0.059_{0.009}$	$\mathbf{0.049}_{0.009}$	$0.058_{0.010}$	$\mathbf{0.044}_{0.011}$
MEND	$0.099_{0.011}$	$0.062_{0.012}$	$0.062_{0.012}$	$0.103_{0.012}$
IKE	$\mathbf{0.005}_{0.000}$	$0.285_{0.011}$	$\mathbf{0.015}_{0.001}$	$0.179_{0.011}$

Table 5: ACE on WikiData$_{counterfact}$.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$-0.141_{0.012}$	$\mathbf{-0.032}_{0.011}$	$0.116_{0.0012}$	$0.104_{0.013}$
ROME	$-0.018_{0.007}$	$-0.104_{0.013}$	$-0.025_{0.007}$	$0.030_{0.015}$
MEND	$-0.009_{0.011}$	$0.072_{0.008}$	$\mathbf{0.003}_{0.011}$	$0.209_{0.012}$
IKE	$\mathbf{-0.003}_{0.000}$	$-0.073_{0.011}$	$-0.018_{0.003}$	$\mathbf{0.010}_{0.013}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.066_{0.003}$	$0.076_{0.004}$	$0.124_{0.005}$	$0.132_{0.006}$
ROME	$-0.055_{0.0003}$	$-0.143_{0.011}$	$-0.064_{0.004}$	$-0.081_{0.012}$
MEND	$-0.061_{0.011}$	$\mathbf{0.066}_{0.011}$	$0.052_{0.008}$	$0.057_{0.005}$
IKE	$\mathbf{-0.004}_{0.003}$	$-0.074_{0.011}$	$\mathbf{0.007}_{0.005}$	$\mathbf{0.032}_{0.012}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$-0.137_{0.012}$	$-0.123_{0.012}$	$\mathbf{-0.005}_{0.012}$	$\mathbf{0.038}_{0.012}$
ROME	$-0.045_{0.004}$	$-0.143_{0.011}$	$-0.063_{0.004}$	$-0.086_{0.012}$
MEND	$-0.083_{0.012}$	$\mathbf{0.034}_{0.011}$	$0.074_{0.013}$	$0.194_{0.013}$
IKE	$\mathbf{-0.011}_{0.001}$	$-0.108_{0.013}$	$-0.018_{0.003}$	$-0.158_{0.013}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$-0.083_{0.011}$	$-0.031_{0.009}$	$\mathbf{-0.039}_{0.011}$	$\mathbf{-0.016}_{0.011}$
ROME	$-0.052_{0.004}$	$-0.157_{0.012}$	$-0.064_{0.004}$	$-0.126_{0.013}$
MEND	$-0.077_{0.010}$	$\mathbf{-0.013}_{0.012}$	$-0.077_{0.011}$	$0.065_{0.012}$
IKE	$\mathbf{-0.016}_{0.001}$	$-0.158_{0.012}$	$-0.065_{0.003}$	$-0.158_{0.012}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$-0.056_{0.011}$	$-0.015_{0.011}$	$0.079_{0.011}$	$0.053_{0.011}$
ROME	$0.048_{0.010}$	$-0.021_{0.012}$	$0.041_{0.012}$	$\mathbf{-0.005}_{0.014}$
MEND	$-0.077_{0.011}$	$\mathbf{-0.011}_{0.012}$	$\mathbf{-0.011}_{0.012}$	$0.088_{0.013}$
IKE	$\mathbf{-0.005}_{0.000}$	$-0.191_{0.011}$	$-0.015_{0.001}$	$-0.128_{0.011}$

Table 6: MCS on WikiData$_{counterfact}$.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.042_{0.006}$	$0.037_{0.005}$	$0.049_{0.006}$	$0.043_{0.006}$
ROME	$0.854_{0.010}$	$0.775_{0.012}$	$0.848_{0.010}$	$0.736_{0.012}$
MEND	$0.846_{0.010}$	$0.808_{0.011}$	$0.8132_{0.010}$	$0.763_{0.011}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.986}_{0.003}$	$\mathbf{1.000}_{0.000}$	$\mathbf{0.917}_{0.008}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.000_{0.000}$	$0.000_{0.000}$	$0.000_{0.000}$	$0.000_{0.000}$
ROME	$0.983_{0.003}$	$0.917_{0.008}$	$\mathbf{0.969}_{0.005}$	$0.798_{0.011}$
MEND	$0.506_{0.013}$	$0.402_{0.013}$	$0.783_{0.012}$	$0.648_{0.013}$
IKE	$\mathbf{0.999}_{0.002}$	$\mathbf{0.993}_{0.002}$	$0.936_{0.007}$	$\mathbf{0.955}_{0.006}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.060_{0.007}$	$0.048_{0.006}$	$0.132_{0.009}$	$0.105_{0.009}$
ROME	$0.962_{0.005}$	$0.879_{0.009}$	$0.942_{0.010}$	$0.859_{0.010}$
MEND	$0.802_{0.011}$	$0.625_{0.013}$	$0.853_{0.011}$	$0.778_{0.011}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.999}_{0.001}$	$\mathbf{0.999}_{0.006}$	$\mathbf{0.957}_{0.006}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.040_{0.005}$	$0.038_{0.006}$	$0.062_{0.007}$	$0.062_{0.007}$
ROME	$0.966_{0.005}$	$0.887_{0.009}$	$\mathbf{0.964}_{0.009}$	$\mathbf{0.878}_{0.009}$
MEND	$0.916_{0.008}$	$0.836_{0.010}$	$0.921_{0.010}$	$0.828_{0.010}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.941}_{0.004}$	$0.961_{0.014}$	$0.417_{0.014}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.064_{0.007}$	$0.067_{0.007}$	$0.095_{0.008}$	$0.090_{0.008}$
ROME	$0.574_{0.014}$	$0.510_{0.014}$	$0.682_{0.013}$	$0.604_{0.014}$
MEND	$0.826_{0.011}$	$0.792_{0.012}$	$0.844_{0.010}$	$0.681_{0.013}$
IKE	$\mathbf{1.000}_{0.000}$	$\mathbf{0.999}_{0.001}$	$\mathbf{0.998}_{0.001}$	$\mathbf{0.952}_{0.005}$

Table 7: Accuracy on ZsRE.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.026_{0.004}$	$0.023_{0.004}$	$0.190_{0.007}$	$0.184_{0.007}$
ROME	$0.078_{0.009}$	$\mathbf{0.071}_{0.009}$	$0.097_{0.009}$	$0.070_{0.009}$
MEND	$0.043_{0.007}$	$0.086_{0.009}$	$0.075_{0.008}$	$\mathbf{0.046}_{0.008}$
IKE	$\mathbf{0.004}_{0.000}$	$0.115_{0.004}$	$\mathbf{0.020}_{0.001}$	$0.110_{0.006}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$\mathbf{0.095}_{0.004}$	$\mathbf{0.082}_{0.003}$	$0.191_{0.005}$	$0.164_{0.005}$
ROME	$0.057_{0.003}$	$0.116_{0.006}$	$0.061_{0.004}$	$0.077_{0.007}$
MEND	$0.117_{0.010}$	$0.090_{0.010}$	$0.105_{0.009}$	$0.074_{0.009}$
IKE	$0.029_{0.001}$	$0.087_{0.003}$	$\mathbf{0.025}_{0.005}$	$\mathbf{0.023}_{0.004}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.031_{0.006}$	$\mathbf{0.038}_{0.006}$	$0.043_{0.007}$	$\mathbf{0.041}_{0.007}$
ROME	$0.052_{0.003}$	$0.091_{0.005}$	$0.038_{0.004}$	$0.075_{0.006}$
MEND	$0.149_{0.010}$	$0.143_{0.011}$	$0.037_{0.006}$	$0.046_{0.007}$
IKE	$\mathbf{0.011}_{0.000}$	$0.070_{0.002}$	$\mathbf{0.006}_{0.001}$	$0.097_{0.005}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$\mathbf{0.016}_{0.004}$	$\mathbf{0.022}_{0.005}$	$\mathbf{0.020}_{0.005}$	$\mathbf{0.012}_{0.004}$
ROME	$0.062_{0.004}$	$0.122_{0.006}$	$0.062_{0.003}$	$0.109_{0.006}$
MEND	$0.039_{0.005}$	$0.083_{0.008}$	$0.047_{0.006}$	$0.065_{0.008}$
IKE	$0.018_{0.001}$	$0.137_{0.005}$	$0.072_{0.004}$	$0.211_{0.013}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.033_{0.006}$	$\mathbf{0.032}_{0.006}$	$0.064_{0.006}$	$0.058_{0.006}$
ROME	$0.078_{0.009}$	$0.074_{0.009}$	$0.092_{0.010}$	$0.071_{0.009}$
MEND	$0.046_{0.007}$	$0.082_{0.008}$	$0.050_{0.008}$	$\mathbf{0.046}_{0.008}$
IKE	$\mathbf{0.006}_{0.000}$	$0.058_{0.002}$	$\mathbf{0.024}_{0.002}$	$0.133_{0.004}$

Table 8: ECE on ZsRE.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.025_{0.005}$	$0.021_{0.004}$	$0.190_{0.007}$	$0.184_{0.007}$
ROME	$0.072_{0.009}$	$\mathbf{0.072}_{0.009}$	$0.086_{0.009}$	$0.061_{0.009}$
MEND	$0.042_{0.008}$	$0.087_{0.009}$	$0.066_{0.008}$	$\mathbf{0.039}_{0.008}$
IKE	$\mathbf{0.004}_{0.000}$	$0.115_{0.004}$	$\mathbf{0.020}_{0.001}$	$0.100_{0.006}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$\mathbf{0.095}_{0.004}$	$\mathbf{0.082}_{0.003}$	$0.191_{0.005}$	$0.164_{0.005}$
ROME	$0.055_{0.003}$	$0.115_{0.006}$	$0.058_{0.004}$	$0.075_{0.008}$
MEND	$0.112_{0.010}$	$0.084_{0.011}$	$0.104_{0.010}$	$0.069_{0.009}$
IKE	$0.029_{0.001}$	$0.086_{0.003}$	$\mathbf{0.025}_{0.005}$	$\mathbf{0.018}_{0.004}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.036_{0.006}$	$\mathbf{0.041}_{0.005}$	$0.043_{0.008}$	$0.043_{0.007}$
ROME	$0.050_{0.004}$	$0.091_{0.005}$	$0.033_{0.004}$	$0.074_{0.007}$
MEND	$0.158_{0.010}$	$0.145_{0.011}$	$0.028_{0.006}$	$\mathbf{0.040}_{0.008}$
IKE	$\mathbf{0.011}_{0.000}$	$0.070_{0.002}$	$\mathbf{0.005}_{0.001}$	$0.096_{0.005}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.021_{0.005}$	$\mathbf{0.020}_{0.005}$	$\mathbf{0.020}_{0.005}$	$\mathbf{0.014}_{0.004}$
ROME	$0.062_{0.004}$	$0.122_{0.006}$	$0.060_{0.004}$	$0.108_{0.006}$
MEND	$0.037_{0.005}$	$0.085_{0.008}$	$0.055_{0.006}$	$0.069_{0.008}$
IKE	$\mathbf{0.018}_{0.001}$	$0.135_{0.004}$	$0.072_{0.004}$	$0.209_{0.011}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$0.035_{0.006}$	$\mathbf{0.035}_{0.006}$	$0.060_{0.006}$	$0.054_{0.006}$
ROME	$0.079_{0.009}$	$0.071_{0.009}$	$0.091_{0.010}$	$0.069_{0.009}$
MEND	$0.045_{0.007}$	$0.083_{0.008}$	$0.052_{0.008}$	$\mathbf{0.045}_{0.008}$
IKE	$\mathbf{0.006}_{0.000}$	$0.057_{0.001}$	$\mathbf{0.023}_{0.002}$	$0.133_{0.005}$

Table 9: ACE on ZsRE.

	Llama-2-7B		Llama-2-7B-chat	
	p	p^*	p	p^*
pre-edit	$0.022_{0.005}$	$0.017_{0.005}$	$0.190_{0.007}$	$0.184_{0.007}$
ROME	$0.062_{0.009}$	$\mathbf{-0.009}_{0.010}$	$0.082_{0.009}$	$0.037_{0.010}$
MEND	$0.007_{0.008}$	$-0.080_{0.009}$	$0.063_{0.008}$	$\mathbf{0.026}_{0.009}$
IKE	$\mathbf{-0.004}_{0.000}$	$-0.115_{0.004}$	$\mathbf{-0.020}_{0.001}$	$-0.099_{0.007}$
	Qwen2.5-7B		Qwen2.5-7B-Instruct	
	p	p^*	p	p^*
pre-edit	$0.095_{0.004}$	$0.082_{0.003}$	$0.191_{0.005}$	$0.164_{0.005}$
ROME	$0.055_{0.003}$	$-0.115_{0.006}$	$-0.058_{0.004}$	$-0.075_{0.008}$
MEND	$-0.090_{0.010}$	$\mathbf{-0.061}_{0.010}$	$-0.071_{0.010}$	$-0.047_{0.010}$
IKE	$\mathbf{-0.029}_{0.001}$	$-0.086_{0.003}$	$\mathbf{0.023}_{0.005}$	$\mathbf{-0.014}_{0.005}$
	Llama3-8B		Llama3-8B-Instruct	
	p	p^*	p	p^*
pre-edit	$\mathbf{0.006}_{0.007}$	$\mathbf{0.012}_{0.006}$	$0.039_{0.008}$	$0.039_{0.007}$
ROME	$-0.050_{0.004}$	$-0.090_{0.005}$	$-0.074_{0.004}$	$-0.074_{0.007}$
MEND	$-0.147_{0.010}$	$-0.137_{0.011}$	$0.025_{0.007}$	$\mathbf{0.020}_{0.008}$
IKE	$-0.011_{0.000}$	$-0.070_{0.002}$	$\mathbf{-0.005}_{0.001}$	$-0.096_{0.005}$
	Llama3.2-3B		Llama3.2-3B-Instruct	
	p	p^*	p	p^*
pre-edit	$\mathbf{0.011}_{0.005}$	$\mathbf{0.008}_{0.005}$	$\mathbf{0.013}_{0.006}$	$\mathbf{0.004}_{0.004}$
ROME	$-0.062_{0.004}$	$-0.122_{0.006}$	$-0.060_{0.004}$	$-0.108_{0.006}$
MEND	$-0.035_{0.005}$	$-0.083_{0.008}$	$-0.040_{0.006}$	$-0.064_{0.008}$
IKE	$-0.018_{0.001}$	$-0.135_{0.004}$	$-0.072_{0.004}$	$0.204_{0.015}$
	Mistral-7B-v0.1		Mistral-7B-Instruct-v0.1	
	p	p^*	p	p^*
pre-edit	$-0.021_{0.006}$	$-0.028_{0.006}$	$0.059_{0.007}$	$0.054_{0.006}$
ROME	$0.0736_{0.010}$	$\mathbf{-0.000}_{0.010}$	$0.064_{0.011}$	$\mathbf{0.027}_{0.011}$
MEND	$-0.033_{0.008}$	$-0.079_{0.009}$	$-0.041_{0.008}$	$-0.034_{0.010}$
IKE	$\mathbf{-0.006}_{0.000}$	$-0.057_{0.001}$	$\mathbf{-0.023}_{0.001}$	$-0.133_{0.005}$

Table 10: MCS on ZsRE.

How Do Large Language Models Evaluate Lexical Complexity?

Abdelhak Kelious[1], **Mathieu Constant**[1], **Christophe Coeur**[2]
[1]University of Lorraine and CNRS/ATILF [2]Consultant
{abdelhak.kelious, mathieu.constant}@univ-lorraine.fr
christophe.coeur@gmail.com

Abstract

In this work, we explore the prediction of lexical complexity by combining supervised approaches and the use of large language models (LLMs). We first evaluate the impact of different prompting strategies (zero-shot, one-shot, and chain-of-thought) on the quality of the predictions, comparing the results with human annotations from the CompLex 2.0 corpus. Our results indicate that LLMs, and in particular gpt-4o, benefit from explicit instructions to better approximate human judgments, although some discrepancies remain. Moreover, a calibration approach to better align LLMs predictions and human judgements based on few manually annotated data appears as a promising solution to improve the reliability of the annotations in a supervised scenario.

1 Introduction

The prediction of lexical complexity is an essential task for adapting linguistic content to the specific needs of learners and educational systems. Such a task consists in predicting a numerical complexity score for a target word in a given sentence (thereafter an instance). Data annotation plays a key role in this task, directly influencing the performance of supervised models. With the emergence of large-scale language models (LLMs) the possibility of using automatically generated annotations raises new questions regarding the generalization and robustness of these models.

In this work, we focus on measuring the similarities between human annotators and generative models (LLMs) by varying the prompts. The objective is to determine whether it is possible to use LLMs as reliable annotators by measuring their level of agreement with human annotations and by analyzing the distribution of the produced annotations. We also seek to identify new perspectives for improving the alignment between these two sources of annotations using a calibration model based on few manually annotated data. We specifically apply this approach in the context of a supervised model trained on LLM-based annotated data to avoid the use of LLMs at prediction time, prioritizing time efficiency and energy conservation. All our experiments were performed on the CompLex 2.0 dataset (Shardlow et al., 2021) for English. This dataset has the advantage of including the source individual human annotations that can be used for directly comparing human and LLM annotations.

The paper is organized as follows. Section 2 presents related work with respect to lexical complexity prediction and data annotation using LLMs. Next, section 3 describes the Complex 2.0 dataset, the LLM strategies to be tested as well as the supervised model used in the final experiments. Then, sections 4 and 5 evaluate LLMs performance against human annotations. Finally, section 6 explores supervised scenarios integrating a calibration model for LLMs.

2 Related work

2.1 Lexical complexity prediction

Lexical complexity is a key issue in text simplification and accessibility. North et al. (2023) provide a comprehensive review of computational methods for predicting lexical complexity primarily in English texts. Their work aims to enhance comprehension by identifying complex words and substituting them with simpler alternatives. The review covers both traditional machine learning techniques, such as support vector machines and logistic regression, and advanced deep neural network models. Moreover, the authors emphasize the use of diverse features including psycholinguistic cues, word frequency, and word length and discuss dedicated competitions, datasets, and practical applications in readability assessment and text simplification across multiple languages.

Research emerging from shared tasks on this sub-

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 348–361
November 8-9, 2025 ©2025 Association for Computational Linguistics

ject highlights the evolution of the field and significant advances in lexical complexity prediction.In 2018, the Complex Word Identification (CWI) shared task marked a turning point by proposing systems capable of identifying words that may be difficult for readers, depending on various contexts. Yimam et al. (2018) revealed that simple models based on n-grams could rival more complex approaches, emphasizing the importance of data and linguistic features in this task. In addition, Gooding and Kochmar (2018) proposed a method based on an ensemble system using majority voting among several models, demonstrating that combining diverse predictors improves overall performance and yields robust results. Furthermore, Kajiwara and Komachi (2018) explored an approach based on lexical frequency in a learner corpus, showing that this methodology is particularly well suited for educational contexts.

Research on predicting lexical complexity has progressed significantly thanks to contributions from the shared task LCP 2021 (Shardlow et al., 2021), which explored the prediction of the complexity of simple words and multiword expressions. Pan et al. (2021) proposed an approach based on a deep ensemble combining pre-trained models such as BERT with advanced techniques such as pseudo-labeling and data augmentation, achieving remarkable results, including first place for multiword expressions. Similarly, Yaseen et al. (2021) used pre-trained models BERT and RoBERTa to compute complexity scores on a continuous scale, ranking first for simple words with a Pearson correlation coefficient of 0.788. Moreover, Mosquera (2021) demonstrated that manual engineering of contextual, lexical and semantic features can still rival modern models, obtaining high correlations for both simple words and multiword expressions. In a more recent study on the LCP 2021 dataset, Kelious et al. (2024b) compared the performance of ChatGPT with that of dedicated models, showing that prompt engineering allows ChatGPT to be competitive, albeit less consistent than specialized models, which reached an R^2 score of 0.65. In parallel, the same authors explored multilingual strategies, comparing supervised and generative approaches to predict lexical complexity. The generative models, although achieving high correlations with prompting strategies (zero-shot, one-shot, etc.), are still surpassed by models optimized for specific tasks. These contributions illustrate a combination of modern and traditional approaches to address the challenges of lexical complexity in both monolingual and multilingual contexts (Kelious et al., 2024a).

Recent research on predicting lexical complexity and text simplification, particularly in multilingual contexts, demonstrates significant advances through the integration of modern techniques. The BEA 2024 shared task explored these aspects in ten languages, using open and proprietary language models, while showing the potential for improvement in complex tasks (Shardlow et al., 2024). Enomoto et al. (2024) [TMU-HIT] demonstrated the effectiveness of GPT-4 in assessing and simplifying lexical complexity in various multilingual contexts, particularly for under-resourced languages, without resorting to supervised data. Similarly, Seneviratne and Suominen (2024) used generative prompts to simplify texts in English and Sinhala, confirming the utility of generative models in less common languages. Another innovative approach used machine translation to predict lexical complexity and simplify texts, combining regressors based on linguistic features with quantized generative models to generate suitable lexical substitutions (Cristea and Nisioi, 2024).

2.2 LLMs for data annotation

Large language models (LLMs) offer significant potential to transform data annotation by reducing costs and increasing efficiency. The work of Liu et al. (2023) presents a systematic review of prompting methods based on LLMs, which allow zero-shot or few-shot learning through structured prompts and pre-trained models, thereby opening up new opportunities for automating annotation. Moreover, Tan et al. (2024) explore how LLMs, such as GPT-4, can generate annotations, classify eligible data types, and address challenges related to bias and annotation quality. Gilardi et al. (2023) show that ChatGPT outperforms human workers in text annotation tasks, with increased accuracy (25 percent higher) and costs 30 times lower. In the field of computational social science, Ziems et al. (2024) demonstrate that while LLMs do not surpass specialized models for classification, they produce qualitative explanations that can enhance research in annotation and creative generation. Other works, such as those by Farr et al. (2024), combine chains of LLMs for more robust and scalable annotation by aggregating predictions from multiple models, while Qiu et al. (2025) use ensembles of LLMs for the evaluation of unstructured textual data, thereby improving annotation consistency. Research by Watts et al. (2024) focuses on the divergences between humans and LLMs for multilingual and multicultural data, highlighting the importance of cultural contexts in annotation. Finally, in software engineering, LLMs show their potential to replace manual annotations, though they remain limited in complex technical contexts (Ahmed et al., 2024). RED-CT, proposed by Farr et al. (2025), illustrates a hybrid approach combining LLM annotations and human interventions for linguistic classification tasks in constrained environments.

3 Data and models

This section will present the models and the data that will be used to (i) evaluate the performances of LLMs with respect to a gold standard and also with respect to individual human annotations; (ii) evaluate the impact of LLMs in a supervised scenario where the LLMs are only used to annotate the training dataset, in order to reduce the energy costs and improve response-time efficiency.

3.1 Dataset

Recently released lexical complexity datasets (Shardlow et al., 2021, 2024) usually provide for each instance a gold numerical complexity score that is the average of several numerical human annotations. In this paper, our goal is to compare LLMs and human annotations. It therefore requires the use of a dataset where all individual human annotations are available, and not only the average of their annotations. This is why for our evaluations, we use the "CompLex 2.0" dataset, an improvement over "CompLex 1.0" (Shardlow et al., 2021). This corpus contains individual human evaluations of the lexical complexity of a set of English texts, carried out using a 5-point Likert scale. The texts included in the corpus come from sources such as Wikipedia, educational books, and newspaper articles, covering a wide variety of topics. The texts were annotated by human evaluators who assessed the lexical complexity of a target word in its context (sentence) using the Likert scale. Each instance was annotated several times, and the average of these annotations was used as the complexity score for each data instance. This score, once normalized, represents a continuous value between 0 and 1. In CompLex 2.0, part of the data from CompLex 1.0 was reused, but the annotations were enriched by adding 10 additional annotations per instance, carried out via the Amazon Mechanical Turk (MTurk) platform, while keeping the same annotation instructions as before. In total, for this second phase, 523 available workers annotated the data, implying that all instances were not annotated by the same workers, which is clearly a limit for the sake of comparison. Furthermore, in the release of Complex 2.0, we only have the data provided by MTurk of the second annotation phase. Therefore, when it comes to comparing with individual human annotations, we will use this data only.

The training and test data contain 7,662 instances and 917 instances respectively.

3.2 LLMs strategies

We used three prompt approaches to evaluate the ability of large language models (LLMs) to predict in-context lexical complexity as proposed by (Kelious et al., 2024a). First, the **Zero-shot prompt (_b)** relies solely on the model's prior knowledge, without providing any specific examples. Next, the **One-shot prompt (_i)** provides a clearer framework by incorporating annotation instructions and a concrete example, allowing the model to better grasp the task at hand. Finally, the **Chain-of-thought prompt (_a)** goes further by exposing detailed instructions, a step-by-step methodology, and an illustrative example to structure the model's reasoning before producing an answer. These three strategies allow the evaluation of complexity from different angles, yielding variable results.The full text of each prompt is provided in the appendix A.

We will experiment with 7 different generative models with the 3 prompts: llama3:8b (Dubey et al., 2024), mistral:7b (Jiang et al., 2023), gemma:9b (Team et al., 2024), phi3:3.8b (Abdin et al., 2024), gpt-4o (January-2025) [1], falcon3:7b (Almazrouei et al., 2023), qwen2:7b (Yang et al., 2024)

Except for gpt-4o, which we access via the OpenAI API, we use the 4-bit quantized versions of all other models and evaluate them with Ollama[2], an open-source tool.

3.3 Supervised model

The supervised scenario consists in using a recent system that has proven effective for predicting lexical complexity in English (Kelious et al., 2024b). The model combines a pre-trained language model with frequency-based features derived from Zipf's law.

---

[1] gpt-4o: `https://openai.com`

[2] ollama : `https://ollama.com`

In summary, the prediction formula is:

$$\hat{y} = f\Big(W_h \cdot \sigma\big(W_e \cdot E + W_f \cdot F + b_e\big) + b_h\Big)$$

where:

- $\hat{y}$ is the predicted complexity value, between 0 and 1;
- E correspond to the lexical embeddings extracted from a transformer model (e.g., DeBerta) from the sequence: [CLS] sentence [SEP] target_word;
- F is the input frequency-based feature vector, $[F_1, F_2, F_3, F_4, F_5]$ [3];
- W_e and W_f are the weights applied respectively to the lexical embeddings (E) and the features (F);
- b_e and b_h are the bias terms for the input layer and the hidden layer;
- σ is a non-linear activation function (ReLU) applied to the combination of E and F;
- W_h corresponds to the weights of the hidden layer;
- f is the linear activation function at the output.

4 Evaluation of LLMs performances against human-based gold complexity scores

In this section, we analyze the performances of the 21 LLM systems derived from our three prompting strategies (section 3.2). We compare the predicted lexical complexity scores with the gold scores, that are, for each instance, the average of several individual human numerical annotations.

4.1 Pearson Correlation Analysis

According to Figure 1, the performance of the models follows a clear trend where the addition of structure and examples improves their ability to predict lexical complexity: on average, the **Zero-shot (_b)** strategy achieves 0.214, the **One-shot (_i)** 0.365, and the **Chain-of-Thought (_a)** [COT] 0.439, confirming the positive impact of explicit reasoning. Comparatively, gpt-4o outperforms all other models, showing high correlations even in Zero-shot (0.746) and reaching 0.780 in COT, while Llama-3 and Mistral show good performance but remain far behind, requiring more advanced prompts to improve their results. In contrast, Phi-3 and Falcon-3 are noticeably less performant, particularly in Zero-shot (respectively 0.023 and 0.088), and need the COT to reach better levels, while Gemma completely fails to capture lexical complexity, with a negative close-to-zero correlation in One-shot (-0.003). In conclusion, the advantage of advanced models like gpt-4o is undeniable, but prompt optimization remains essential to improve the performance of weaker models.

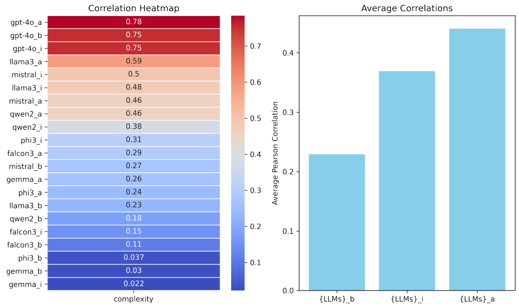


Figure 1: Pearson correlation between complexity predicted by LLMs and the gold complexity (left part); and average correlation (right part) with respect to the prompt strategy types (_b: zero-shot, _i: one-shot, _a: chain-of-thoughts).

4.2 Predicted complexity and error distributions

The violin plot in Figure 2 provides a more detailed view of the distribution of the model predictions compared to the distribution of human-based gold complexity scores on the test set. Figure 4 in Appendix B provides a complementary view showing the distributions of the residuals, i.e. the LLM errors (y_{gold} - y_{llm}).

Distribution of gold complexity scores (complexity): The distribution of values is quite spread out, meaning that the perception of lexical complexity by human annotators varies according to the instances. There is a notable concentration around specific values, which may indicate that most words have a moderately perceived complexity (neither too easy nor too difficult). Some extreme values exist, which could correspond to words that are widely considered either very simple or very complex.

Models close to gold annotations: The models gpt-4o (gpt-4o_b, gpt-4o_i, gpt-4o_a) and Llama3 (llama3_i, llama3_a) display distributions similar to human complexity. Their medians are relatively

---

[3] F1 (the Zipf score of the word frequency), F2 (the average Zipf score in a sentence), F3 (the difference between the target word's Zipf score and the average score), F4 (the number of words with a Zipf score higher than the target word) and F5 (a binary value indicating whether the target word is considered rare with a score less than or equal to 3).

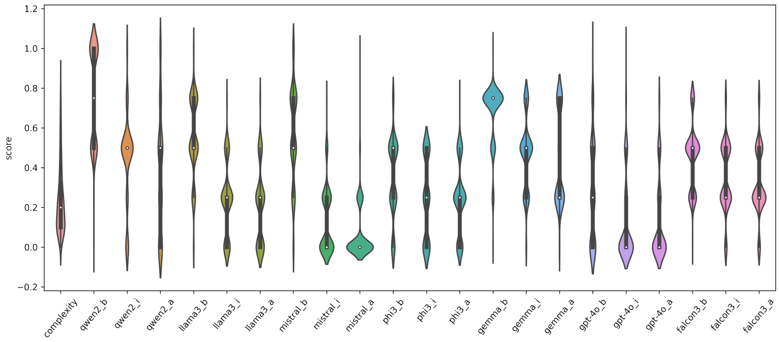


Figure 2: Distribution of lexical complexity predictions for each LLM and distribution of gold scores ("complexity" violin plot)

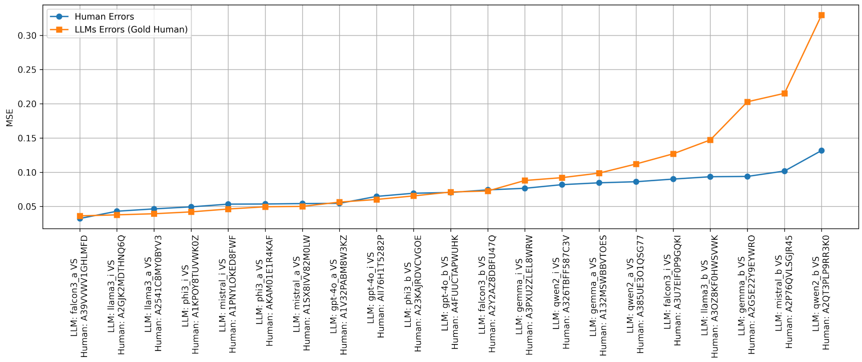


Figure 3: Comparison between individual human annotations and LLMs, by selecting the top-21 annotators with lowest MSE.

aligned with the gold annotations and their predictions cover a comparable range of values, indicating a certain consistency. Mistral (mistral_i, mistral_a) follows a similar trend with moderate dispersion, suggesting that it evaluates lexical complexity in a balanced manner, without excessively overestimating or underestimating. These trends are confirmed with the error distributions.

Models with notable discrepancies: Some models show more marked divergences compared to human annotations. Phi3 (phi3_b, phi3_i, phi3_a) and Qwen2 (qwen2_b, qwen2_i, qwen2_a) have a higher median, indicating a tendency to overestimate word complexity. Falcon3_i and Falcon3_a, on the other hand, display a larger dispersion, notably Falcon3_i which sometimes assigns exceptionally high values. This variability suggests a lack of stability in the predictions, which can be problematic for reliable evaluation of lexical complexity.

General insights and implications: The models gpt-4o, Llama3, and Mistral seem to be the closest to human annotations, suggesting that they could be the most reliable for predicting lexical complexity. Conversely, Phi3 and Falcon3 tend to overestimate complexity, while Qwen2 shows more rigid predictions biased toward higher values. A high dispersion in predictions, as observed in some models, may indicate inconsistency or a lack of calibration, while an overly concentrated distribution can reflect a lack of diversity in the evaluation of words. Thus, if the goal is to mimic human perception, the models most aligned with the annotations should be favored, whereas those with high variability or marked bias might require adjustment for better calibration.

5 Comparison of LLM predictions with individual human annotations

In this section, we compare LLM predictions with the individual human annotations. Unfortunately, the way the dataset is annotated using the MTurk platform with a limitation of 10 human annotations per instance makes a fair comparison difficult, whereas we have the predictions of all the LLMs per instance. Although there are clear limitations in the various provided evaluations below due to this issue, the results will reveal some trends that will pave the way for other experiments.

5.1 Comparing the LLMs with the best human annotators

Since we have 7 LLMs and 3 different prompts, making a total of 21 models, we will compare this set with the top 21 human annotators selected based on their Mean Square Error (MSE) score. Note that we did not use the Pearson correlation scores as the difference of two Pearson correlation scores is difficult to interpret with two different sets of annotated instances (all instances for LLMs vs. various numbers of instances for human annotators).

General Error Comparison (MSE): Figure 3 shows that human annotators generally have lower errors than LLMs. Indeed, most annotators display a more stable and homogeneous MSE, whereas LLMs show much more variability in their performance. Some models come close to human performance, while others have much larger discrepancies.

Error Dispersion: Human errors range between approximately 0.03 and 0.13 for the top-21, indicating a certain consistency in their annotations. In contrast, LLM errors are much more dispersed, ranging from 0.03 up to over 0.32, suggesting significant heterogeneity depending on the model used. Some LLMs are very performant, while others clearly struggle to reproduce precise annotations.

Best and Worst Performers: The best models and annotators are those that display the lowest MSE. Among the LLMs, Falcon3_a (MSE $\approx$ 0.036) and Llama3_i (MSE $\approx$ 0.038) stand out for their precision, rivaling the best human annotators, notably "A39VVWV1GHLMFD" (MSE $\approx$ 0.032) and "A2GJK2MDTHNQ6Q" (MSE $\approx$ 0.043). Conversely, some models display particularly high errors. Qwen2_b (MSE $\approx$ 0.33) is the least precise among the LLMs, followed by Mistral_b (MSE $\approx$ 0.21). On the human side, "A2QT3PLP9RR3K0" is the annotator whose annotations deviate the most from the reference values (MSE $\approx$ 0.13).

Direct Comparison between LLMs and Human Annotators: Some LLMs manage to achieve, or even surpass, the performance of the least precise human annotators among the top-21 ones. The graph shows that up to the 10th-best annotator there is more or less an equivalence between human and LLM performances.

5.2 Comparing LLMs with individual human annotations on a common set of instances

In the ideal case, comparing LLMs with individual human annotations should be performed on a common set of instances. To make the analysis manageable due to impractical combinatorics in CompLex 2.0 to find the set of annotators with the largest set of shared annotated instances, we chose to take the five annotators who annotated the largest number of instances and extract the 375 instances annotated in common. This approach reduces the scope of the problem while retaining a representative set of annotations for our analyses.

On this subset of instances, we performed an evaluation using standard evaluation metrics (R^2, Pearson Coefficient, and MSE) comparing annotators and LLMs. For each of the annotators (annot1..5), we selected the five LLM/humans whose evaluations were in the closest agreement with theirs (according to Cohen's Quadratic Kappa metric). Table 1 provides the results of the evaluation metrics.

Human	Model	R^2	Pearson	MSE	Kappa
annot1, (MSE :0.021)	gpt-4o_a	**0.4801**	**0.6929**	**0.0345**	**0.68**
	gpt-4o_i	0.4194	0.6476	0.0410	0.62
	gpt-4o_b	0.4116	0.6415	0.0518	0.60
	annot3	0.3398	0.5830	0.0610	0.57
	annot5	0.3288	0.5734	0.0542	0.57
annot2, (MSE :0.039)	annot1	**0.1612**	**0.4015**	0.0810	**0.36**
	annot3	0.1448	0.3806	0.0957	0.36
	gpt-4o_b	0.1338	0.3659	**0.0798**	0.36
	annot5	0.1481	0.3848	0.0988	0.33
	gpt-4o_a	0.0901	0.3001	0.1002	0.24
annot3, (MSE :0.034)	annot1	**0.3398**	**0.5830**	**0.0610**	**0.57**
	annot5	0.3076	0.5546	0.0722	0.54
	gpt-4o_b	0.2927	0.5410	0.0728	0.53
	gpt-4o_a	0.2982	0.5461	0.0688	0.50
	gpt-4o_i	0.2741	0.5236	0.0747	0.47
annot4, (MSE :0.039)	gpt-4o_b	**0.0529**	**0.2300**	0.0883	**0.21**
	llama3_a	0.0446	0.2113	**0.0655**	0.21
	annot2	0.0461	0.2147	0.0878	0.19
	annot1	0.0319	0.1787	0.0775	0.18
	mistral_i	0.0224	0.1497	0.0728	0.16
annot5, (MSE :0.030)	annot1	**0.3288**	**0.5734**	0.0542	**0.57**
	annot3	0.3076	0.5546	0.0722	0.54
	gpt-4o_a	0.2978	0.5457	**0.0537**	0.54
	gpt-4o_i	0.2777	0.5270	0.0562	0.51
	gpt-4o_b	0.2404	0.4903	0.0830	0.44

Table 1: Results of evaluation metrics (R^2, Pearson, MSE, Kappa) comparing annotators annot1...5 and models. The 5-closest annotators annot1...5 or LLM models are provided for each human annotator annot1...5 with respect to Cohen's Quadratic Kappa (Kappa).

Overall, we can see that for each selected human annotator there are three LLMs in its 5-closest humans/LLMs (exception: only two LLMs for annot2). It shows that we can always find an LLM closer to her/him than two other human annotators (only one for annot2). The gpt4o LLMs tend to be the closest to the selected human annotators: 3 occurrences in the top-5 for three human annotators (annot1, annot3 and annot5), 2 occurrences for annot2 and only one occurrence (zero-shot) for annot4, the latter emerging as an "outlier" (low correlation with everyone).

This view is of course partial because of the specificity of the selected human annotators (the ones who annotated the largest number of instances) that are not representative of all annotators. This should be investigated further by enlarging the set of annotators (but reducing the evaluation set), and/or by varying selection criteria in order to have more global view. Nevertheless, the preliminary investigation presented in this section show some potential for aligning individual human annotators and LLMs.

6 Train supervised model

In a real scenario, annotating an instance using 10 LLMs simultaneously would be very expensive in terms of time, money and energy cost compared to using a small supervised model. In this section, we train various supervised models on the CompLex 2.0 dataset, trying to take advantage of LLMs to annotate the training data, and therefore limiting their use to an offline setting.

6.1 Preliminary cross-evaluation

We first perform evaluations using the supervised model described in section 3.3, notably crossing the various types of annotations available. In particular, the Complex 2.0 dataset contains, for each instance, individual annotations from Amazon Mechanical Turk as well as an overall score that incorporates other inaccessible annotations. We distinguish three types of annotations:

- **llms**: the average of the annotations provided by several language models (LLMs). To simulate the Amazon Mechanical Turk approach, we randomly select 10 LLMs out of 21, recalling that MTurk selects 10 annotators from among 523.
- **mturk**: the average of the scores assigned by the human annotators from Amazon Mechanical Turk.
- **all**: the average of all annotations, that is, those from MTurk plus the additional inaccessible annotations (global score).

Train → Test	**Pearson**	$\mathbf{R^2}$	**MSE**
all → all	0.79	0.62	0.0065
mturk → mturk	0.87	0.76	0.0072
llms → llms	0.78	0.62	0.0080
all → mturk	0.86	0.74	0.0100
all → llms	0.50	0.25	0.1780
mturk → all	0.79	0.63	0.1210
mturk → llms	0.53	0.28	0.3320
llms → mturk	0.57	0.33	0.0450
llms → all	0.52	0.27	0.0250

Table 2: Results of Pearson, R^2, and MSE for each *train → test* setting.

Intra-ensemble performance (homogeneous): When both training and testing are performed on annotations of the same type, the performance is high (Table 2). For instance, the scenario *all → all* (r = 0.79, R^2 = 0.62, MSE = 0.0065) illustrates good consistency when human annotators are used for both training and testing. Similarly, the *mturk → mturk* approach (r = 0.87, R^2 = 0.76, MSE = 0.0072) gives the highest results, reflecting the high homogeneity of MTurk annotators. Finally, in *llms → llms* (r = 0.78, R^2 = 0.62, MSE = 0.0080), the language models generate annotations that are globally consistent with each other, even though they remain slightly below the quality obtained with MTurk.

Cross-performance (heterogeneous): In a context where training and testing come from different sources, the generalization varies greatly. The *all → mturk* approach (Pearson = 0.86, R^2 = 0.74, MSE = 0.010) shows a fairly good capacity of the model to predict the MTurk-specific annotations when trained on data annotated by a larger set of human annotator. Conversely, *all → llms* (Pearson = 0.50, R^2 = 0.25, MSE = 0.178) results in a significant drop in performance, revealing a marked divergence between the annotations generated by LLMs and those by humans. The *mturk → all* option (Pearson = 0.79, R^2 = 0.63, MSE = 0.121) remains relatively satisfactory, but the increase in MSE indicates a difficulty in fully capturing the diversity of the annotations. Finally, *mturk → llms* (Pearson = 0.53, R^2 = 0.28, MSE = 0.332) confirms a notable incompatibility between the judgments of MTurk and those of the generative models.

Impact of LLMs with respect to human annotations: When training on annotations from LLMs to test on MTurk (*llms → mturk*), the performance remains modest (r = 0.57, R^2 = 0.33, MSE = 0.045), demonstrating that the models do not fully capture the complexity as perceived by human annotators. Similarly, the scenario *llms → all* (r = 0.52, R^2 = 0.27, MSE = 0.025) yields similar results:

LLMs do not faithfully reproduce the judgments from a mixed set of human annotations.

6.2 Calibrating LLMs

The results in the previous section indicate that, despite their internal consistency, LLMs require significant adjustments to align their annotations with human judgments, especially in subjective tasks such as lexical complexity prediction. To do so, we propose the following three-step method: (1) we train a calibration model on N samples from the training set to learn how to combine the predictions from the various LLMs, (2) we directly apply this model to generate annotations on the training set; and (3) we train a supervised model (section 3.3) on these *pseudo-labels* and evaluate it on the test set produced by human annotators.

The proposed calibration model combines LLMs using a weighting scheme that can be mathematically formulated as:

$$\hat{y} = \sum_{i=1}^{n} \alpha_i \cdot x_i + b$$

where n is the number of LLMs (21 in our case) and α_i is the weight associated to the complexity score x_i predicted by the LLM LLM_i, the term b being the bias. The weights and the bias are trained by minimizing the MSE on the subset of training data annotated by humans.

Sample size (N)	**Pearson**	**R^2**	**MSE**
100	0.73	0.54	0.0145
500	0.75	0.55	0.0108
1 000	0.74	0.55	0.0103
2 000	0.74	0.55	0.0114
5 000	0.75	0.57	0.009
All (7 662)	0.74	0.56	0.0116
No weights (avg)	0.44	-2.90	0.0635
Model llms → all	0.52	0.27	0.0250
Model all → all	0.79	0.62	0.0065

Table 3: Evaluation of the LLMs calibration on test set.

Table 3 provides the performances of the supervised model based on the calibration model predictions to annotate the training dataset, varying the sample size N. It appears that with only few annotated data (N=100) we can observe a significant improvement of the performances with respect to using a simple average of the LLMs predictions to annotate the training set: Pearson increases from 0.44 to 0.73, MSE decreases from 0.064 to 0.015. Varying the sample size N from 100 to all instances, the performances remain mostly stable despite some little variations indicating that a N value between 100 and 500 seems sufficient to approach the results of the supervised model (all → all) which remains superior (Pearson=0.79, $R^2 = 0.62$, MSE=0.0065). Note that applying the calibration model directly on the test set yields similar trends as shown in Appendix C, confirming the validity of the approach. Moreover, the condensed error distribution around 0 for the calibrated model applied directly on the test set shows the improved alignment with human annotations (cf. "stacked_calibrated" violin plot in Appendix B). It is also interesting to note that using to simple average method with no weights tend to be better by randomly sampling ten LLMs per instance than by using the all set of LLMs (Table 3).

7 Conclusion

In this study, we explored the prediction of lexical complexity by using large language models (LLMs) with different prompting strategies (zero-shot, one-shot, chain-of-thought). Our experiments show that adding structure and explicit examples significantly improves the models' ability to approach human judgments, with gpt-4o notably standing out with high correlations and better alignment with the reference annotations.

The comparative analysis of predictions distributions and errors (MSE) highlights significant variability between LLM predictions and human evaluations. While some models (such as Llama3 and Mistral) manage to approach human performance in certain scenarios, others (such as Qwen2) exhibit marked biases or excessive dispersion in their predictions. These findings underscore the importance of precise calibration and prompt optimization to fully leverage the capabilities of generative models.

Moreover, although training a supervised model on human annotations remains the performance benchmark (Pearson=0,79, $R^2 = 0,62$, MSE=0,0065), our results show that the use of a calibration model which integrates an optimized weighting of the LLMs' predictions yields significantly higher scores than simply averaging the LLMs predictions, with Pearson coefficients reaching up to 0,75 and R^2 values of 0,57 with as few as 500 examples. This improvement, consistent across various subsets, confirms that calibration by stacking enables a better use of the combined richness of human annotations and automatic predictions, while drastically reducing the number of human annotations required.

In brief, our work shows the potential of LLMs and in particular that of the calibration models for lexical complexity prediction. However, the variability observed in certain metrics, such as the MSE, and the persistent gaps with human annotations call for continued optimization efforts, notably by refining prompting techniques and calibration strategies. Future research could focus on improving the self calibration of generative models and adapting these approaches to other languages and educational contexts, in order to fully exploit the synergy between human annotations and automatic predictions.

8 Limitations

Despite the promising results presented in this study, several limitations must be acknowledged:

- **Choice and size of language models:**
  The analysis was based on a limited set of models (e.g., llama3, Mistral, Gemma, Phi3, gpt-4o, Falcon3, and Qwen2), whose sizes and architectures were chosen based on practical criteria (notably the use of 4-bit quantized versions). Although this selection represents a certain segment of current LLMs, it limits the generalizability of the results. Future investigations could incorporate a greater variety of models and examine the impact of model size and parameter settings on predicting lexical complexity.

- **Focus on English and the Complex 2.0 dataset:**
  This study is limited to the analysis of English texts, relying exclusively on the Complex 2.0 dataset, which was chosen for the richness of its annotations. However, lexical complexity is a phenomenon that can vary significantly across languages due to structural and lexical differences. Extending the analysis to other languages, accompanied by language-specific prompts and guidelines, would help capture intercultural dynamics more accurately and broaden the scope of the conclusions.

- **Simulation of MTurk annotations:**
  The dataset used is based on annotations from 523 participants via Amazon Mechanical Turk. Accurately reproducing this level of heterogeneity is challenging, as simulating the equivalent of 523 annotators using LLMs is difficult. In this study, we limited our analysis to a subset of 5 annotators who annotated the highest number of common instances. Increasing this number in future research would allow for a better estimation of the variability and robustness of human judgments.

- **Calibration method:**
  Although the calibration method has shown its effectiveness in aligning LLM predictions with human annotations, it is only a starting point. A more comprehensive benchmark incorporating various calibration methods would be beneficial in identifying the optimal strategy and further improving the alignment between automatic predictions and human judgments.

- **Data leakage and lexical-complexity biases:**
  Sentences from CompLex 2.0 may already be present in the vast corpora used to pretrain many LLMs, creating potential data leakage that can artificially inflate reported performance. Moreover, *we observed that prediction errors increase markedly* whenever a sentence contains rare vocabulary or syntactically/semantically demanding contexts, revealing a systematic bias toward high-frequency, “easier” language. Together, these issues undermine the reliability of the evaluation and underscore the need for stricter corpus filtering and for benchmarks that better cover the long tail of lexical difficulty.

These limitations pave the way for interesting future work, including extending the analysis to other languages, exploring a greater diversity of models and calibration methods, and incorporating a larger number of annotators to enhance the robustness and generalizability of the results.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Toufique Ahmed, Premkumar Devanbu, Christoph Treude, and Michael Pradel. 2024. Can llms replace manual annotation of software engineering artifacts? *arXiv preprint arXiv:2408.05534*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Petru Cristea and Sergiu Nisioi. 2024. Machine translation for lexical complexity prediction and lexical simplification. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 610–617.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Taisei Enomoto, Hwichan Kim, Tosho Hirasawa, Yoshinari Nagai, Ayako Sato, Kyotaro Nakajima, and Mamoru Komachi. 2024. Tmu-hit at mlsp 2024: How well can gpt-4 tackle multilingual lexical simplification? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 590–598.

David Farr, Nico Manzonelli, Iain Cruickshank, Kate Starbird, and Jevin West. 2024. Llm chain ensembles for scalable and accurate data annotation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 2110–2118. IEEE.

David Farr, Nico Manzonelli, Iain Cruickshank, and Jevin West. 2025. Red-ct: A systems design methodology for using llm-labeled data to train and deploy edge linguistic classifiers. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 58–67.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Sian Gooding and Ekaterina Kochmar. 2018. CAMB at CWI shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194, New Orleans, Louisiana. Association for Computational Linguistics.

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

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex word identification based on frequency in a learner corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, New Orleans, Louisiana. Association for Computational Linguistics.

Abdelhak Kelious, Mathieu Constant, and Christophe Coeur. 2024a. Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches. In *Swedish Language Technology Conference and NLP4CALL*, pages 96–114.

Abdelhak Kelious, Matthieu Constant, and Christophe Coeur. 2024b. Complex word identification: A comparative study between chatgpt and a dedicated model for this task. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3645–3653.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Alejandro Mosquera. 2021. Alejandro mosquera at semeval-2021 task 1: Exploring sentence and word features for lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 554–559.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584.

Jiaxing Qiu, Dongliang Guo, Papini Natalie, Peace Noelle, Levinson Cheri, and Teague R Henry. 2025. Ensemble of large language models for curated labeling and rating of free-text data. *arXiv preprint arXiv:2501.08413*.

Sandaru Seneviratne and Hanna Suominen. 2024. Anu at mlsp-2024: Prompt-based lexical simplification for english and sinhala. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 599–604.

Matthew Shardlow, Fernando Alva-Manchego, Riza Theresa Batista-Navarro, Stefan Bott, Saul Calderon-Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, and Anna Huelsing. 2024. The bea 2024 shared task on the multilingual lexical simplification pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 571–589.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. *arXiv preprint arXiv:2106.00473*.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. Pariksha: A large-scale investigation of human-llm evaluator agreement on multilingual and multi-cultural data. *arXiv preprint arXiv:2406.15053*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. Just-blue at semeval-2021 task 1: Predicting lexical complexity using bert and roberta pre-trained language models. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*, pages 661–666.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

A Prompts

1- Zero-shot prompt (base)

`"""`

`You will be given a sentence and a word included in the sentence. Evaluate the complexity of the word in the context of the sentence, and provide a rating in scale of 0.0, 0.25, 0.5, 0.75, 1.0.`

`Sentence: '{sentence}'`

`Word: '{token}'`

`Complexity:`

`return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word in context.`

`"""`

2- One-shot prompt (instruct)

`"""`

`You are a person without specialized knowledge or expertise in any specific field.You will receive a sentence containing a word, your task is to evaluate the word based on one metric.`

`Evaluation Criteria:`

`Complexity [0.0, 0.25, 0.5, 0.75, 1.0]: This measures how difficult it is to understand the word.`

`1. Carefully examine the sentence and the specified word to grasp the context in which it is used.`

`2. Assess the complexity of the word using the criteria provided`

`- 0.0: The word is simple and easily understandable to most people.`

`- 0.25: The word may have some complexity or be specific to a certain field, but can still be understood with some effort.`

`- 0.5: The word is moderately complex and may require some background knowledge or explanation to understand fully.`

`- 0.75: The word is quite complex and may be difficult to understand without significant knowledge or explanation.`

`- 1.0: The word is extremely complex and likely only understood by experts or individuals with specialized knowledge.`

`Your personal knowledge of a word should not influence your rating. Instead, rate the word based` on the understanding an average person might have

Example:

Sentence: 'The professor's discourse was filled with intricate terminology that baffled the students.' Word: 'discourse'.

For this example, 'discourse' might be rated as 0.25.

Please provide a complexity rating for the '{language}'word '{token}'.

Sentence: '{sentence}'

Word: '{token}'

return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word.

"""

3- Chain-of-thought prompt (Advanced Cot)

"""

You are a person without specialized knowledge or expertise in any specific field.You will receive a sentence containing a word, your task is to evaluate the word based on one metric.

Evaluation Criteria:

Complexity [0.0, 0.25, 0.5, 0.75, 1.0]: This measures how difficult it is to understand the word.

Evaluation steps:

- **1.Understand the Context:** - Read the sentence and the word carefully to understand the context in which the word is used.
- **2. Analyze the Word's Frequency and Familiarity:** - Determine how commonly the word is used in everyday language. - Consider if the word is generally known by the average person or if it is specialized.
- **3. Evaluate the Morphological Complexity:** - Examine the structure of the word, including its length, composition, and any prefixes or suffixes.
- **4. Define the Word:** - Provide a definition of the word in its common usage. - Explain the specific meaning of the word in the given context.
- **5. Assess the Overall Complexity:** - Based on the analyses above, determine the complexity of the word using the following criteria: - 0.0: The word is simple and easily understandable to most people. - 0.25: The word may have some complexity or be specific to a certain field, but can still be understood with some effort. - 0.5: The word is moderately complex and may require some background knowledge or explanation to understand fully. - 0.75: The word is quite complex and may be difficult to understand without significant knowledge or explanation. - 1.0: The word is extremely complex and likely only understood by experts or individuals with specialized knowledge.
- **6. Assign a Complexity Rating:** - Based on your evaluation, assign a complexity rating to the word.

Your personal knowledge of a word should not influence your rating. Instead, rate the word based on the understanding an average person might have

Example:

Sentence: 'The professor's discourse was filled with

```
intricate terminology that baffled the students.' Word: 'discourse'

1. Understand the Context: The word 'discourse' is used in a sentence discussing a professor's speech.

2. Analyze the Word's Frequency and Familiarity: 'Discourse' is somewhat specialized but can be understood by most people with some effort.

3. Evaluate the Morphological Complexity: 'Discourse' is a relatively long word but does not have complex prefixes or suffixes.

4. Define the Word: - Common usage: 'Discourse' means written or spoken communication. - Context-specific: In the sentence, 'discourse' refers to the professor's lecture.

5. Assess the Overall Complexity: Considering its moderate frequency, moderate morphological complexity, and clear context-specific meaning, 'discourse' might be rated as 0.25.

6. Assign a Complexity Rating: For this example, 'discourse' might be rated as 0.25.

Now, Please provide a complexity rating for the '{language}'word '{token}'.

Sentence: '{sentence}'

Word: '{token}'

return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word.

"""
```

B Distributions of prediction errors of the LLMs

Figure 4 provides a complementary view showing the distributions of the residuals, i.e. the LLM errors (y_{gold} - y_{llm}).

C Performances of calibrated LLMs

Table 4 shows the performances of the system combining LLMs using the calibration model on the test set.

Sample size (N)	**Pearson**	r^2	**MSE**
100	0.77	0.44	0.0169
500	0.81	0.60	0.0119
1,000	0.81	0.61	0.0118
2,000	0.82	0.64	0.0108
5,000	0.82	0.67	0.0098
All (7,662)	**0.83**	**0.68**	0.0095
No weights (avg)	0.44	-2.9	0.0635
model (all → all)	0.79	0.62	**0.0065**

Table 4: Performance metrics by sample size, applying the calibration model directly to the test set.

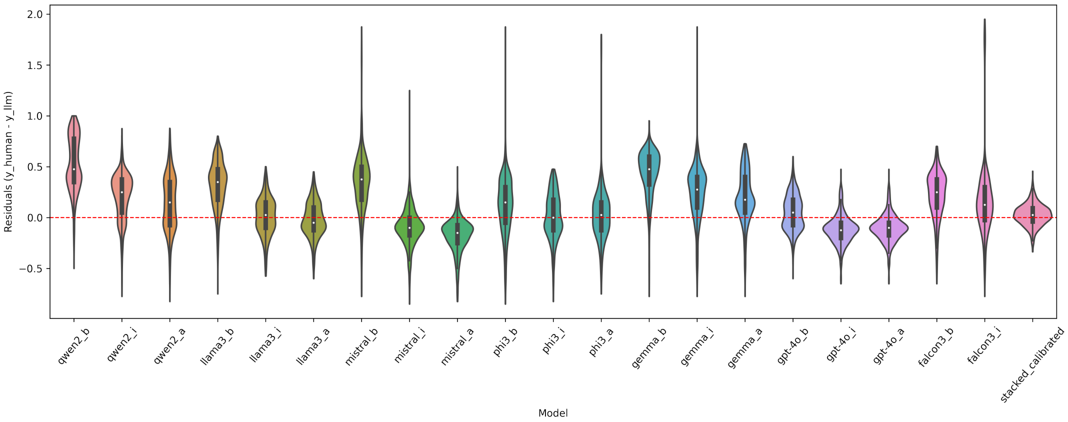


Figure 4: Distribution of errors for each LLM according to gold scores

SAG: Enhancing Domain-Specific Information Retrieval with Semantic-Augmented Graphs

Carol-Luca Gasan and **Vasile Păiș**
Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy
`gasancarolluca@gmail.com, vasile@racai.ro`

Abstract

Retrieval-Augmented Generation (RAG) systems rely on high-quality embeddings to retrieve relevant context for large language models. This paper introduces the Semantic-Augmented Graph (SAG), a new architecture that improves domain-specific embeddings by capturing hierarchical semantic relationships between text segments. Inspired by human information processing, SAG organizes content from general to specific concepts using a graph-based structure. By combining static embeddings with dynamic semantic graphs, it generates context-aware representations that reflect both lexical and conceptual links. Experiments on text similarity and domain-specific question answering show that SAG consistently outperforms standard embedding methods within RAG pipelines.

1 Introduction

Retrieval-Augmented Generation (RAG) enhances large language models by integrating relevant external information during response generation. This process relies on embedding models to convert text into vector representations for efficient similarity-based retrieval. While general-purpose embeddings perform well across many domains, they often fall short in specialized fields like medicine, law, or technical writing, where semantic structures are more complex and hierarchical. To address this gap, we propose the Semantic-Augmented Graph (SAG), a novel architecture designed to improve domain-specific embeddings by modeling how humans organize knowledge. SAG converts unstructured text into a directed graph, with nodes representing semantic units and edges encoding contextual relationships. These graphs are further optimized into subtrees that reflect domain-relevant knowledge clusters, capturing both broad concepts and detailed information.

SAG is particularly suited for environments with limited computational resources, where large-scale models are impractical. By enhancing static embeddings with graph-based structure, SAG achieves an effective balance between efficiency and representational power. The approach aligns with ongoing efforts to reduce the energy and compute costs of NLP systems, offering a scalable solution for domain-adapted retrieval and reasoning.

The rest of the paper is structured as follows: Section 2 reviews related work on embeddings and RAG methods. Section 3 outlines the conceptual foundation of our approach. Section 4 details the SAG architecture and processing pipeline. Section 5 presents experimental results on semantic similarity and question answering. Section 6 concludes with future directions.

2 Related Work

Similarity search in high-dimensional vector spaces is a core component of information retrieval and knowledge discovery. Faiss (Johnson et al., 2019), developed by Meta's AI research team, is a widely adopted library that enables scalable similarity search and clustering. It supports both CPU and GPU execution and implements efficient algorithms using techniques like product quantization and inverted indexing. Faiss allows rapid retrieval from large datasets while supporting multiple distance metrics, making it a foundational tool for retrieval-augmented systems.

Learning effective graph representations is essential for tasks such as node classification, link prediction, and clustering. Node2Vec (Grover and Leskovec, 2016) addresses this by generating node embeddings through biased random walks that capture various neighborhood structures. By preserving network proximity in the embedding space, Node2Vec generalizes earlier methods and provides reliable performance across multiple graph-based applications.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 362–371
November 8-9, 2025 ©2025 Association for Computational Linguistics

Evaluating retrieval-augmented generation systems in specialized domains like medicine requires robust benchmarks. MIRAGE (Rajpurkar et al., 2020) offers a comprehensive suite of medical question-answering datasets and an evaluation framework designed to reflect realistic use cases. Some of them are BioASQ (Tsatsaronis et al., 2015), MedMCQA (Pal et al., 2022), MedQA-USMLE (Marro et al., 2023), and MMLU-MED (Hendrycks et al., 2021). Recent advances in biomedical retrieval have leveraged contrastive learning to improve semantic representations. MedCPT (Yuan et al., 2022) introduces a transformer model trained with contrastive objectives to support zero-shot retrieval in the biomedical domain. It achieves state-of-the-art results on standard benchmarks by generating robust sentence embeddings even in the absence of annotated training pairs.

Knowledge graphs provide structured representations of domain-specific information, supporting complex reasoning and query answering. In the medical field, the traditional static knowledge graphs are being replaced by adaptive systems that incorporate large language models and domain-specific retrieval tools. These systems automatically extract entities, infer relationships, and build query-specific graphs. Frameworks like AMG-RAG (Wang et al., 2023) use these adaptive knowledge graphs in retrieval-augmented pipelines, combining evidence retrieval and reasoning to improve accuracy and interpretability in medical question answering.

3 Model's Philosophy

SAG is inspired by the hierarchical structure of human cognition, where knowledge is processed from general principles to specific details. This structure is especially prominent in expert domains, where reasoning involves traversing semantic layers—from broad categories to precise instances. We define information specificity as "the range of knowledge from which a sentence is derivable". General statements apply broadly, while specific ones are valid in narrower contexts. Consider this spectrum from the biomedical domain:

- General: “Medications are substances used to treat medical conditions.”
- Intermediate: “Anticoagulants prevent blood clot formation.”
- Specific: “Warfarin inhibits vitamin K epoxide reductase, preventing activation of clotting factors II, VII, IX, and X.”

SAG operationalizes this cognitive structure as a directed graph: nodes represent text chunks, and edges encode general-to-specific relationships. This graph captures domain-specific knowledge hierarchies, which we leverage to produce semantically rich embeddings tailored for RAG systems.

4 SAG's Pipeline

4.1 Data and Static Word Embeddings

We begin with unstructured domain-specific text, segmented into semantically coherent chunks (hereafter called paragraphs). Tokenization is handled using a standard English tokenizer with lowercasing; we found no substantial benefit from domain-specific tokenizers in our setup. Initial embeddings are drawn from pre-trained models such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), serving as a static foundation. SAG then fine-tunes these representations through graph-based enrichment. While domain-specific embeddings can improve convergence speed, our results show that SAG's performance is ultimately robust to the choice of initialization.

4.2 Generating initial graph

We aim to construct a directed graph where nodes represent text paragraphs, and edges represent semantic relationships between them. Initially, we create a fully connected graph where each node corresponds to a paragraph. For every pair of nodes, we calculate a similarity score using cosine similarity between their static embeddings. Edges are only preserved if their similarity exceeds a threshold τ (empirically set to 0.5 in our experiments). The edge weight is computed as:

$$w_{ij} = \frac{(\cos(\vec{p_i}, \vec{p_j}) + 1)/2 - \tau}{1 - \tau} \quad (1)$$

where $\vec{p_i}$ and $\vec{p_j}$ are the paragraph embeddings, and the weight is mapped to the range $[0, 1]$.

To compute paragraph embeddings, we use the Smooth Inverse Frequency (SIF) weighting scheme. SIF reduces the influence of common words and emphasizes the importance of informative words in the paragraph. The paragraph embedding $\vec{p_i}$ is computed as:

$$\vec{p_i} = \frac{1}{|T_i|} \sum_{t \in T_i} \frac{a}{a + f(t)} \cdot \vec{t} \quad (2)$$

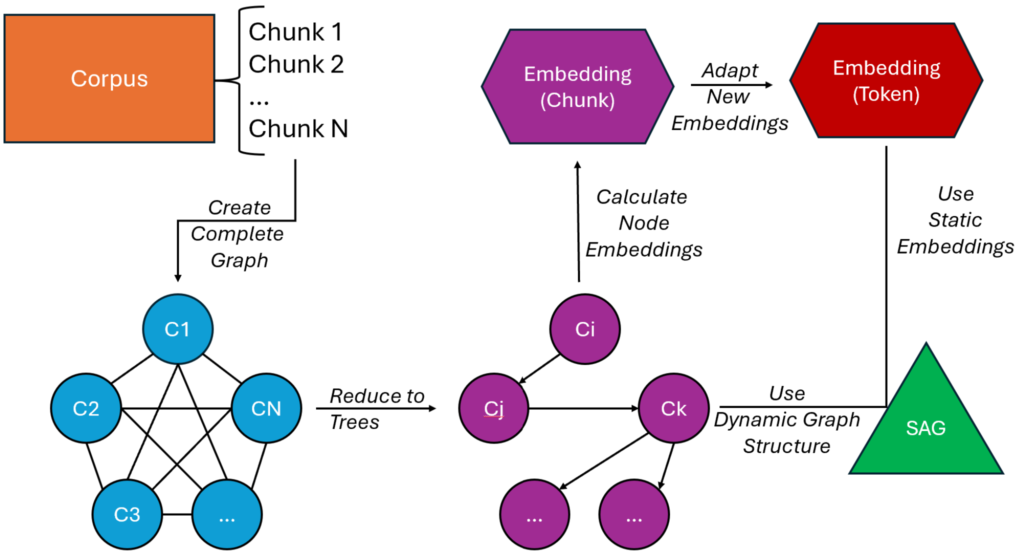


Figure 1: Overview on SAG's pipeline

where T_i is the set of tokens in paragraph i, a is a smoothing parameter (empirically set to 0.0002), $f(t)$ is the frequency of token t in the corpus, and $\vec{t}$ is the token embedding.

In our experimental design, we deliberately selected static word embeddings augmented with SIF for paragraph representations, despite the availability of various sentence and text embedding alternatives. This approach enables us to isolate and highlight the architectural benefits of SAG itself, rather than potentially obscuring its contributions by applying it to already optimized state-of-the-art models, which would make it difficult to distinguish between innate model performance and SAG-derived improvements.

To transform this undirected similarity graph into a directed graph that captures the general-to-specific relationship, we use the SMOG readability index (McLaughlin, 1969), which is particularly effective for medical texts. The SMOG index provides a measure of textual complexity, which we use as a proxy for specificity: higher SMOG scores indicate more specific content, while lower scores suggest more general content. The SMOG index is calculated as:

$$\text{SMOG} = 1.043\sqrt{N_p\left(\frac{30}{N_s}\right)} + 3.1291 \qquad (3)$$

where N_p is the number of polysyllabic words (words with three or more syllables) and N_s is the number of sentences.

For each pair of connected nodes (i, j), we compute their respective SMOG scores s_i and s_j, and derive a directionality measure:

$$c_i = \frac{s_i}{s_i + s_j}, \quad c_j = \frac{s_j}{s_i + s_j} = 1 - c_i \qquad (4)$$

We then establish a directed edge from the node with the lower c value (more general) to the node with the higher c value (more specific). This process transforms our similarity graph into a directed graph that represents the flow from general to specific information.

4.3 Reducing the graph in size

The fully connected directed graph, while theoretically comprehensive, is computationally inefficient and contains redundant information. Our goal is to reduce this graph to a more manageable structure while preserving the most important semantic relationships. Specifically, we aim to transform the graph into a forest of directed trees, where each tree represents a coherent subdomain of knowledge. We employ a heuristic algorithm that combines topological sorting with dynamic programming to identify optimal subtrees, formally described in Algorithm 1. The distinction between trees naturally emerges from the semantic structure of the data, without requiring explicit domain labels. Each tree captures a semantic hierarchy from general to specific information within its subdomain.

Algorithm 1 Graph Reduction to Forest

```
procedure ReduceToForest($G = (V, E)$)
    Initialize $visited[node] \leftarrow 0$ for all $node \in V$
    Initialize $stack \leftarrow []$
    function DFS($node$)
        $visited[node] \leftarrow 1$
        for each neighbor $v$ of $node$ in $E$ do
            if $visited[v] = 0$ then
                DFS($v$)
            end if
        end for
        $stack$.append($node$)
    end function
    for each $node \in V$ do
        if $visited[node] = 0$ then
            DFS($node$)
        end if
    end for
    Reverse $stack$
    Initialize $costs[node] \leftarrow \infty$ for all $node$
    Initialize $oset \leftarrow$ sorted list of $(\infty, node, [])$ for all $node$
    $new\_edges \leftarrow []$
    while $oset$ not empty do
        $(node, curr\_cost, edges) \leftarrow$ pop node with minimum cost from $oset$
        if $curr\_cost = \infty$ then
            $curr\_cost \leftarrow 0$
        end if
        $costs[node] \leftarrow curr\_cost$
        for each $(prev\_node, edge\_cost)$ in $edges$ do
            $new\_edges$.append($(prev\_node, node, \{weight : 1 - edge\_cost\})$)
        end for
        for each outgoing edge $(node, neighbor, weight)$ in $E$ do
            if $neighbor$ is still in $oset$ then
                if $curr\_cost + weight <$ current cost of $neighbor$ then
                    Update $neighbor$'s cost to $curr\_cost + weight$
                    Record path from $node$
                else if $curr\_cost + weight =$ current cost of $neighbor$ then
                    Add alternative path from $node$
                end if
            end if
        end for
    end while
    Replace $G$'s edges with $new\_edges$
    return $G$
end procedure
```

4.4 Computing updated word embeddings

With the optimized forest structure in place, we now compute enhanced node embeddings that incorporate both the textual content and the graph structure. We use Node2Vec to generate embeddings for each node in the forest, capturing the structural relationships between paragraphs. For individual tokens, we compute enriched embeddings by taking a weighted average of the node embeddings for all paragraphs in which the token appears:

$$\vec{t}_{\text{new}} = \sum_{i \in P(t)} \frac{f_{t,i}}{\sum_{j \in P(t)} f_{t,j}} \cdot \vec{n}_i \tag{5}$$

where $P(t)$ is the set of paragraphs containing token t, $f_{t,i}$ is the frequency of token t in paragraph i, and $\vec{n}_i$ is the node embedding for paragraph i.

Rather than completely replacing the original static embeddings, we merge them with these new embeddings to preserve general semantic information while incorporating domain-specific relationships. The combined embeddings are computed as:

$$\vec{t}_{\text{final}} = [\vec{t}_{\text{static}}; \vec{t}_{\text{new}}] \tag{6}$$

After merging the static and new embeddings, the resulting dimensionality increases. To ensure that the final embeddings match the original static embeddings' dimensionality (for seamless integration with existing systems), we apply dimensionality reduction techniques. Specifically, we use the method described in (Raunak et al., 2019) to reduce the size of the word embeddings. The implementation can be seen in Algorithm 2.

Algorithm 2 Dimensionality Reduction via PCA

```
Input: Data matrix $X_{\text{train}}$, components $O$, reduced dims $N$, desired comps $D$, words $W$
Output: Reduced embeddings $\vec{t}_{\text{final}}$
Center data: $X_{\text{train}} \leftarrow X_{\text{train}} - \text{mean}(X_{\text{train}})$
$U_1 \leftarrow \text{PCA}(X_{\text{train}}, O)$    ▷ First PCA: extract main components
for each $x \in X_{\text{train}}$ do
    $x \leftarrow x - \sum_{j=1}^{D} \text{dot}(U_1[j]^T, x) \cdot U_1[j]$    ▷ Remove top D components
end for
$X_{\text{train}} \leftarrow X_{\text{train}} - \text{mean}(X_{\text{train}})$
$X_{\text{new}} \leftarrow \text{PCA}(X_{\text{train}}, N)$    ▷ Project to N dimensions
$X_{\text{new}} \leftarrow X_{\text{new}} - \text{mean}(X_{\text{new}})$
$X_{\text{new}} \leftarrow \text{PCA}(X_{\text{new}}, N)$    ▷ Second PCA refinement
for each $i \in 1..W$ do
    $\vec{t}_{\text{final}}[i] \leftarrow X_{\text{new}}[i] - \sum_{j=1}^{D} \text{dot}(U_{\text{fit}}[j]^T, X_{\text{new}}[i]) \cdot U_{\text{fit}}[j]$
end for
Return $\vec{t}_{\text{final}}$
```

4.5 Encoding logic

4.5.1 Single-Tree

The SAG encoding process integrates static and dynamic components to produce embeddings that capture both lexical content and structural semantics. The static component uses a weighted average of token embeddings with SIF weighting to downplay common terms. The dynamic component incorporates domain-specific relationships by simulating a walk through the semantic graph, as described in Algorithm 3. The final embedding is a weighted combination of the two:

$$\vec{e} = p_{\text{SAG}} \cdot \vec{d} + (1 - p_{\text{SAG}}) \cdot \vec{s} \tag{7}$$

where p_{SAG} (typically 0.7) balances the influence of the graph-based and lexical components.

4.5.2 Multiple-Trees

In practice, our graph reduction algorithm typically produces multiple trees, each representing a distinct subdomain within the corpus. For each text to be encoded, we identify the most relevant trees (termed "master trees") based on token overlap. Specifically, for a given text, we identify the top k trees (where $k = \sqrt{|trees|}$ by default) with the highest token frequency overlap.

When comparing two texts, we compute similarity scores for each tree in the union of their respective master tree sets, with trees that appear in both sets receiving double weight. The final similarity score is:

$$\text{sim}(A, B) = \frac{\sum_{t \in M_A \cup M_B} w_t \cdot \cos(\vec{e}_A^{\,t}, \vec{e}_B^{\,t})}{\sum_{t \in M_A \cup M_B} w_t} \tag{8}$$

where M_A and M_B are the sets of master trees for texts A and B, $\vec{e}_A^{\,t}$ and $\vec{e}_B^{\,t}$ are the embeddings of A and B with respect to tree t, and $w_t = 2$ if $t \in M_A \cap M_B$ and $w_t = 1$ otherwise.

Algorithm 3 Dynamic Embedding Computation

```
procedure ComputeDynamicEmbedding($text, G, nodeEmb, tokenFreq$)
    $tokens \leftarrow$ tokenize($text$)
    $span \leftarrow tokens[0 : \text{SPAN\_SIZE}], \quad last \leftarrow \text{SPAN\_SIZE}$
    $sumVec \leftarrow \sum_{t \in span} tokenFreq[t]$
    $currNode \leftarrow \arg\max_n sumVec[n], \quad vecs \leftarrow [nodeEmb[currNode]]$
    $temp \leftarrow T_0$                                        ▷ Initial temperature
    while $last < |tokens|$ do
        Update $span$ and $sumVec$ with $tokens[last]$
        $last \leftarrow last + 1$
        $steps \leftarrow$ ComputeSteps($span, tokenFreq$), $\quad tmp \leftarrow []$
        for $1 \ldots steps$ do
            With prob. $\propto e^{-1/temp}$, pick random neighbor of $currNode$
            Otherwise, set $currNode \leftarrow \arg\max$ neighbor by $sumVec$
            $temp \leftarrow temp \cdot \lambda$
            $tmp$.append($nodeEmb[currNode]$)
        end for
        $vecs$.append($tmp$)
    end while
    return mean($vecs$)
end procedure
```

4.6 Inference high-level methods

While the multiple-tree approach provides rich semantic representations, it requires a custom similarity function, which may not integrate seamlessly with existing embedding-based tools and pipelines. To address this, we propose a compact embedding format that encapsulates the multi-tree information within a single vector representation, facilitating compatibility with standard similarity metrics.

4.6.1 Compact Embedding Formulation

Let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be the set of all trees in our forest, and $\mathcal{M}_x \subset \mathcal{T}$ be the set of master trees for text x. The compact embedding $\vec{c}_x$ is constructed as a concatenation of three components:

$$\vec{c}_x = [\vec{s}_x^{\,\text{weighted}}; \vec{e}_x^{\,\text{specific}}; \vec{e}_x^{\,\text{general}}] \tag{9}$$

where:

- $\vec{s}_x^{\,\text{weighted}} = r_{\text{sent}} \cdot \vec{s}_x$ represents the static component weighted by r_{sent}
- $\vec{e}_x^{\,\text{specific}} = [p_{\text{SAG}} \cdot \vec{e}_x^{\,i} \text{ if } T_i \in \mathcal{M}_x \text{ else } \vec{0}]_{i=1}^n$ represents the tree-specific embeddings for master trees
- $\vec{e}_x^{\,\text{general}} = [p_{\text{SAG}} \cdot \vec{e}_x^{\,i}]_{i=1}^n$ represents the general tree embeddings across all trees

The weighting factor r_{sent} is computed as:

$$r_{\text{sent}} = \begin{cases} 2 \cdot |\mathcal{M}_x| \cdot p_{\text{sent}} & \text{if } |\mathcal{M}_x| > 0 \\ p_{\text{sent}} & \text{otherwise} \end{cases} \tag{10}$$

where $p_{\text{sent}} = 1 - p_{\text{SAG}}$.

4.6.2 Theoretical Justification

The compact embedding design ensures that when comparing two texts using standard cosine similarity, the result approximates our weighted multi-tree similarity. Consider the dot product between compact embeddings of texts x and y:

$$\begin{aligned} \vec{c}_x \cdot \vec{c}_y = {} & r_{\text{sent}}^2 \, (\vec{s}_x \cdot \vec{s}_y) \\ & + p_{\text{SAG}}^2 \sum_{i=1}^{n} \left(1 + I_{T_i \in \mathcal{M}_x \cap \mathcal{M}_y}\right) (\vec{e}_x^{\,i} \cdot \vec{e}_y^{\,i}) \end{aligned} \tag{11}$$

where $I(\cdot)$ is the indicator function and $I(A)I(B) = I(A \cap B)$ was used to merge the two sums.

When normalized by the magnitudes in the cosine similarity calculation, this approximates our weighted multi-tree similarity Formula 8. The compact embedding computation is implemented efficiently as shown in Algorithm 4.

Algorithm 4 Compact Embedding Computation

```
procedure ENCODE($text, p_{SAG} = 0.7, inference = True$)
    $p_{sent} \leftarrow 1 - p_{SAG}$
    $sent \leftarrow$ ComputeStaticEmbedding($text$)
    $sent \leftarrow \frac{sent}{||sent||}$                                  ▷ Normalize
    $masters, sag \leftarrow$ ComputeSAGVectors($text$)
    $r_{sent} \leftarrow 2.0 \cdot |masters| \cdot p_{sent}$ if $|masters| > 0$ else $p_{sent}$
    $sentPart \leftarrow [r_{sent} \cdot sent]$
    $specificPart \leftarrow [p_{SAG} \cdot sag[i]$ if $i \in masters$ else $\vec{0}$ for $i$ in range($|sag|$)]
    $generalPart \leftarrow [p_{SAG} \cdot x$ for $x$ in $sag]$
    if inference then
        $parts \leftarrow [sentPart, specificPart, generalPart]$
    else
        $parts \leftarrow [sentPart, generalPart, specificPart]$
    end if
    $result \leftarrow []$
    for $part \in parts$ do
        for $vector \in part$ do
            $result$.extend($vector$.flatten())
        end for
    end for
    return array($result$)
end procedure
```

4.7 Parameter Discussion

Parameters fall into training and inference categories. Training's key factors are chunk size and graph node count (controlling granularity/scale); similarity threshold mainly reduces computation by limiting edges without affecting structure. For inference, embedding weight between static/dynamic components is crucial and tunable per use case. The 50-dim embeddings balance performance with speed, aligning with typical subtree counts (6-10); larger sizes showed comparable accuracy but slower inference.

5 Experiments

We evaluated SAG on text similarity and RAG-based question answering tasks. To optimize training data quality, we utilized the CRAFT 2.0 dataset (Cohen et al., 2017) containing 67 full-text articles with around 560,000 tokens. Our training employed 1024 chunks of 512 tokens each with final static embeddings of size 50.

5.1 Text Similarity

We evaluated SAG on two biomedical semantic textual similarity datasets: Clinical STS EBMSASS (Hassanzadeh et al., 2019), containing 1,000 expert-annotated clinical evidence pairs with 1-5 similarity scores from biomedical abstracts, and BIOSSES (Sogancioglu et al., 2017), comprising 100 sentence pairs from biomedical articles with 0-4 similarity scores focused on citation relationships. Evaluation employed a domain-specific pretrained Sentence Transformer trained on the PubMed corpus (Wheeler et al., 2002). Table 4 demonstrates SAG's state-of-the-art performance, exceeding both the reference model and previous SOTA on BIOSSES, and marginally outperforming the reference model on EBMSASS.

5.2 Question Answering with RAG

We evaluated SAG on the MIRAGE benchmark while considering two retrieval corpora: LoData (medical textbooks and Statpearls (Publishing, 2025)) and HiData (25% of PubMed). Table 2 demonstrates that RAG augmented with SAG consistently outperforms baseline approaches using Gemini 2.0 Flash. Notably, Table 5 indicates SAG's minimal computational overhead, with reduced inference time and memory requirements in both single-threaded and accelerated environments, confirming its suitability for resource-constrained deployments. SAG enhances standard RAG pipelines to competitive levels with special-

Model	Size	Acc. (%)	Chain-of-Thought	Web-Search
Med-Gemini (Saab et al., 2024)	1800B	91.1	Yes	Yes
GPT-4 (OpenAI, 2023)	1760B	90.2	Yes	Yes
Gemini 2.0 Flash + RAG w/ SAG (HiData)	40B	87.7	Yes	No
Med-PaLM 2 (Singhal et al., 2025)	340B	85.4	Yes	No
Med-PaLM 2 (5-shot) (Singhal et al., 2025)	340B	79.7	Yes	No
AMG-RAG (Wang et al., 2023)	8B	73.9	Yes	Yes
Meerkat (Kim et al., 2024)	7B	74.3	Yes	No
Meditron (Chen et al., 2023)	70B	70.2	Yes	Yes
Flan-PaLM (Singhal et al., 2023)	540B	67.6	Yes	No
LLAMA-2 (Chen et al., 2023)	70B	61.5	Yes	No
Shakti-LLM (Shakhadri et al., 2024)	2.5B	60.3	No	No
BioMedGPT (Luo et al., 2023)	10B	50.4	No	No

Table 1: Recent results on MedQA-US.

Model	BioASQ	MedMCQA	MedQA-US	MMLU-MED	Avg
RAG with SAG (HiData)	74.13	71.88	87.65	90.16	80.95
RAG with MedCPT (HiData)	66.55	69.43	84.23	85.71	76.48
RAG with SAG (LoData)	72.35	67.85	67.45	83.49	72.79
RAG with MedCPT (LoData)	65.12	66.73	65.42	79.96	69.31
No RAG	60.35	64.50	67.01	78.84	67.68

Table 2: Accuracy (%) on Question Answering Datasets using Gemini 2.0 Flash

Model	Acc.
SAG (HiData)	71.88
AMG-RAG (Wang et al., 2023)	66.34
Meditron (Chen et al., 2023)	66.00
Codex 5-shot (Liévin et al., 2024)	59.70
VOD (Liévin et al., 2023)	58.30
PubmedBERT (Gu et al., 2021)	40.00

Table 3: Recent results on MedMCQA

Dataset	SAG	Reference	SOTA
EBMSASS	85.84	85.62	–
BIOSSES	96.87	86.67	93.63

Table 4: Pearson Correlation (%) on biomedical STS datasets

Model	Size	Time S/A	Memory S/A
SAG	124 MB	136/32 ms	1/7 KB
MedCPT	439 MB	350/62 ms	7/23 MB

Table 5: Inference Time and Memory Usage per 1k tokens in (S)ingle-threaded and (A)ccelerated settings

ized medical models, achieving 87.7% on MedQA-USMLE (Table 1) and 71.88% on MedMCQA (Table 3). These results demonstrate SAG's efficacy as a lightweight enhancement module enabling modest LLMs to compete with state-of-the-art systems in complex biomedical reasoning.

6 Conclusion and Future Work

This work presents the Semantic-Augmented Graph (SAG), a new architecture that improves domain-specific embeddings by organizing semantic relationships hierarchically within a directed graph. By structuring information from general to specific concepts, SAG produces context-aware embeddings while remaining more computationally efficient than traditional deep learning methods. Experimental results show that SAG achieves state-of-the-art performance on biomedical semantic similarity benchmarks. When integrated into a Retrieval-Augmented Generation system, it consistently improves biomedical question-answering performance while retaining its efficiency. Future research may explore expanding SAG to general-domain applications, adapting it for symbolic and mathematical data, combining multiple embedding sources through ensemble methods, and extending the model to low-resource languages where efficient learning is critical. We anticipate releasing the code and related artifacts in the future to support reproducibility and further research.

7 Limitations

Despite the promising results, our study is subject to several limitations. First, due to hardware constraints when processing large-scale graphs with Python-based backend libraries, the Semantic-Augmented Graph (SAG) was restricted to a maximum of 1024 nodes during training and inference. Our implementation using NetworkX for graph operations on million-edge structures faced memory and processing bottlenecks, despite running on dual Intel Xeon Silver 4210 CPUs (2.20GHz, 40 total logical cores) and two NVIDIA GPUs (Quadro RTX 6000/8000 and RTX 5000). While the node limitation was sufficient for capturing meaningful hierarchical structures, it may have prevented deeper modeling of larger corpora. Second, during RAG evaluation, only a quarter of the full PubMed dataset was accessible, which may have constrained the retrieval coverage and affected downstream performance. These limitations highlight potential gains from scaling SAG with more efficient graph processing backends and accessing larger retrieval corpora. As with any embedding-based system applied in biomedical contexts, there is a potential risk of encoding latent biases or producing misleading similarity scores that may impact downstream clinical decisions. We recommend caution and further evaluation before deployment in real-world healthcare settings.

8 Acknowledgements

This research was in part sponsored by the NATO Science for Peace and Security Programme under grant id. G8648.

References

Mark Chen, Khaled Saab, Harsha Nori, Aakanksha Chowdhery, Tiffany Kuo, Jure Lee, Joon Lee, Yifan Zhang, Qian Jin, and Raj Patel. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.00000*.

K. Bretonnel Cohen, Karin Verspoor, Karën Fort, Christopher Funk, Michael Bada, Martha Palmer, and Lawrence E. Hunter. 2017. The colorado richly annotated full text (craft) corpus: Multi-model annotation in the biomedical domain. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 1379–1394. Springer Netherlands, Dordrecht.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Pubmedbert: A domain-specific language model for biomedical text. *arXiv preprint arXiv:2007.15779*.

Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2019. Ebmsass: Evidence-based medicine sentence similarity dataset. In *Proceedings of the 10th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 652–657.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Zou, Dawn Lee, Dawn Tang, Dawn Song, Jacob Steinhardt, Justin Gilmer, Erica Ma, Gaurav Sastry, Andy Tran, Xander Wang, Andy Miller, Alexander D'Amour, Andrew Lohn, David Krueger, and 23 others. 2021. Measuring massive multitask language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23711–23723.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Jinhyuk Kim, Wonjin Kim, Jinhyuk Lee, Joongbo Lee, Kyunghyun Lee, Sunghwan Yoon, and Jaewoo Kang. 2024. Meerkat: A medical reasoning benchmark for large language models. *arXiv preprint arXiv:2402.00000*.

Valentin Liévin, Charlotte E Hother, and Ole Winther. 2023. Vod: Visual open-domain question answering. *arXiv preprint arXiv:2305.00000*.

Valentin Liévin, Charlotte E Hother, and Ole Winther. 2024. Can large language models reason about medical questions? *npj Digital Medicine*.

Ruijie Luo, Yuan Li, Yuxuan He, Yuxian Wang, Xiaodan Zhang, Shijie Wang, Shuo Zhang, Xiaozhi Liu, Zhiyuan Liu, and Maosong Sun. 2023. Biomedgpt: A unified and generalist biomedical generative pretrained transformer for biomedical text, image, and cross-modal tasks. *arXiv preprint arXiv:2306.00000*.

Stefano Marro, Benjamin Molinet, Elena Cabrio, and Serena Villata. 2023. Natural language explanations for clinical case retrieval. In *Artificial Intelligence in Medicine*. Springer.

G. Harry McLaughlin. 1969. Smog grading: A new readability formula. *Journal of Reading*, 12:639–646.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*.

OpenAI. 2023. Gpt-4 technical report. ArXiv preprint arXiv:2303.08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

StatPearls Publishing. 2025. Statpearls [internet]. Online medical reference, regularly updated. Available from: https://www.statpearls.com/.

Pranav Rajpurkar, Emily Chen, Bridget McCall, Raj Patel, Jenny Liu, Jeremy Irvin, James Zou, Ross Jones, Nikhil Kohli, Tony Duan, and Daisy Ding. 2020. Mirage: A large-scale medical qa benchmark for robustness and generalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Khaled Saab, Harsha Nori, Aakanksha Chowdhery, Tiffany Kuo, Jure Lee, Joon Lee, Yifan Zhang, Qian Jin, Raj Patel, and Percy Liang. 2024. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*.

A Shakhadri, S Sahoo, S Saha, A Kumar, S Kumar, A Gupta, and A Sharma. 2024. Shakti-llm: An open-source large language model for indian languages. *arXiv preprint arXiv:2403.00000*.

Karan Singhal, Shekoofeh Azizi, Tong Tu, Soroush Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Hunter Cole, Joon Lee, and Zachary Bradshaw. 2023. Large language models encode clinical knowledge. *Nature*. Doi:10.1038/s41586-023-05881-4.

Karan Singhal, Shekoofeh Azizi, Tong Tu, Soroush Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Hunter Cole, Joon Lee, and Zachary Bradshaw. 2025. Toward expert-level medical question answering with large language models. *Nature*. Doi:10.1038/s41586-024-07366-5.

Gizem Sogancioglu, Hakime Öztürk, and Arzucan Özgür. 2017. Biosses: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, John Pavlopoulos, Theofanis Karaletsos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Niko Heino, Eric Gaussier, Laura Barrio-Alvers, and 2 others. 2015. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. *Journal of Biomedical Informatics*, 57:1–7.

Yifan Wang, Linjun Zhang, Yichong Liu, Yichao Zhang, Jimmy Lin, Buzhou Tang, Xiaodan Wang, Hua Xu, Fei Wang, and Yulan He. 2023. Amg-rag: Adaptive medical graphs for retrieval-augmented generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

David L. Wheeler, Deanna M. Church, Scott Federhen, Alex E. Lash, Thomas L. Madden, Joan U. Pontius, Gregory D. Schuler, Lynn M. Schriml, Eduardo Sequeira, Tatiana A. Tatusova, and Lisbeth Wagner. 2002. Pubmed: the bibliographic database. *Nucleic Acids Research*, 30(1):61–65.

Hongyin Yuan, Kai Sun, Qingyu Yu, Buzhou Tang, Xiaolong Wang, Bin Wang, Xiaodan Wang, Yanshan Zhang, Hua Xu, Fei Wang, Yulan He, and Fei Wang. 2022. Medcpt: Contrastive pre-training for biomedical retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Cross-Domain Persuasion Detection with Argumentative Features

Bagyasree Sudharsan and **Maria Leonor Pacheco**
University of Colorado Boulder
`{bagyasree.sudharsan, maria.pacheco}@colorado.edu`

Abstract

The main challenge in cross-domain persuasion detection lies in the vast differences in vocabulary observed across different outlets and contexts. Superficially, an argument made on social media will not look like an opinion presented in the Supreme Court, but some of the latent factors that make an argument persuasive are common across all settings. Regardless of domain, persuasive arguments tend to use sound reasoning and present solid evidence, build on the credibility and authority of the source, or appeal to the emotions and beliefs of the audience. In this paper, we show that simply encoding the different argumentative components and their semantic types can significantly improve a language model's ability to detect persuasion across vastly different domains.

1 Introduction

Persuasion is the process of guiding someone to adopt a particular way of thinking or behaving. It is a natural part of everyday life, helping people resolve small disagreements and find common ground. At the same time, it is also a powerful tool used by leaders and institutions to shape society's understanding of broader and more complex issues. Different arguments have different levels of persuasiveness, often determined by specific syntactic and semantic markers (Habernal and Gurevych, 2016a; Ta et al., 2022). Interestingly, this holds true regardless of the context in which arguments are made. In Fig. 1 we present two examples from two different domains (Reddit and Supreme Court proceedings) where the same argumentative strategy was used with the intent of persuading. In both cases, the writer introduced an interpretative claim and offered a premise that appealed to the emotions and beliefs of the audience (known as pathos). In this paper, we build on the idea that these strategies can be applied across domains and use them to detect persuasion in unseen language contexts.

> [Welcoming immigrants and refugees has been our country's unfair advantage]$_{claim\_interpretation}$... [As many of you know, I am the son of an undocumented immigrant from Germany and the great grandson of refugees who fled the Armenian Genocide]$_{premise\_pathos}$

> [Under the Due Process and Equal Protection Clauses of the 14th Ammendment couples of the same-sex may not be deprived of that right and that liberty]$_{claim\_interpretation}$... [Their hope is not to be condemned to live in loneliness, excluded from one of civilization's oldest institutions. They ask for equal dignity in the eyes of the law]$_{premise\_pathos}$

Figure 1: Similar argumentation strategies in two domains. Reddit post by Alexis Ohanian, co-founder of Reddit (Top). Supreme Court opinion by Justice Anthony Kennedy in Obergefell vs. Hodges (Bottom).

The task of detecting persuasion is not new; numerous studies have examined this problem, primarily with the aim of identifying persuasive strategies to counter misinformation and propaganda (Da San Martino et al., 2020; Nikolaidis et al., 2024). Much of this research has aimed to identify specific techniques in persuasive material with varying degrees of success. Previous studies have examined diverse media, including news articles (Piskorski et al., 2023), social media posts (Tan et al., 2016), legal proceedings (Danescu-Niculescu-Mizil et al., 2012), images (Liu et al., 2023), and memes (Dimitrov et al., 2024). They have also addressed a wide range of domains, from political discourse (Lazer et al., 2018) to medical information (Kamali et al., 2024). However, the vast majority of these studies remain confined to a single domain.

In this paper, we study persuasion detection in a cross-domain setting. Prior work has looked at some aspects of cross-domain transfer in persuasion-related tasks, such as topic-agnostic persuasive dialogue generation (Jin et al., 2024) and cross-lingual variation in persuasive language (Li

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 372–380
November 8-9, 2025 ©2025 Association for Computational Linguistics

et al., 2024). In the former work, although topics varied, all the conversations followed the same general structure and style; a turn-taking discussion about daily-life situations. In the latter work, although the language varied, all instances were taken from the same media platform. In contrast, our work focuses on exploring whether explicitly modeling argumentation components and their modes of persuasion can improve cross-domain transfer when domains differ along more than one dimension. We build on the premise that argumentation strategies are somewhat domain-agnostic and look at three domains with different purposes, audiences, structures, and argumentation styles. To this end, we make the following contributions.

1. We propose a simple framework for introducing information about the structure of the argument and the persuasion mode for the persuasion detection task.
2. We design a challenging cross-domain experiment using three domains that differ significantly in language use, argument length, and argumentation style: social media, legal proceedings, and formal debates on general topics.
3. We show that regardless of this variation, argument information can significantly improve the ability of fine-tuned language models to detect persuasion across domains.

2 Related Work

Persuasion Detection. There are two main lines of work in the space of persuasion detection: one that frames the problem as a classification task in which the goal is to identify if a text instance is (more or less) persuasive (Habernal and Gurevych, 2016a,b; Dutta et al., 2020; Darnoto et al., 2023), and one that is concerned with identifying the specific persuasion techniques employed in the text (Braca and Dondio, 2023; Dimitrov et al., 2021, 2024; Iyer and Sycara, 2019; Nayak and Kosseim, 2024). Most of these studies employ fully supervised approaches with training data from the target domain. A notable exception is the framework proposed by Iyer and Sycara (2019), which forgoes supervision by relying on syntactic parse trees to identify persuasion tactics. In contrast, we rely on simple signals from the argumentation structure to achieve cross-domain transfer.

Argumentation Mining. There is a large body of work dealing with argumentation mining in the context of persuasive texts. Some studies focus on extracting trees from long documents to represent the overall structure of the arguments made (Stab and Gurevych, 2017; Widmoser et al., 2021). Earlier work has also used argument components to improve high-level persuasion detection (Dutta et al., 2020), although in single-domain scenarios. Chakrabarty et al. (2019) combines these two areas of work by first extracting structured arguments using rhetorical structure theory and then infusing this information into a language model to identify persuasive online discussions. We follow a similar idea but considerably simplify the way in which we model the argumentation structure.

3 Methodology

In this section, we present the argumentation taxonomy and datasets used, as well as our approach to predict argumentation components and persuasion.

3.1 Argumentation Taxonomy

We build on the taxonomy used by Hidey et al. (2017) to qualify opinions on Reddit. They use two types of argumentation components: *claims* and *premises*. Claims are the main statements or conclusions that are being proven. They represent the key ideas that the writer wants the audience to accept as true. Premises, on the other hand, are the supporting statements that provide the reasons or evidence for the claims. The premises serve as the foundation on which the arguments are built.

Premises are further classified into three semantic types according to their mode of persuasion. These modes include *ethos* (appeals to the writer's character), *logos* (appeals to reason), and *pathos* (appeals to emotions and beliefs). We also consider different combinations of these semantic types. For claims, the types are based on a simplification of the Freeman proposition (Freeman, 2011) - *interpretation*, *rational evaluation*, *emotional evaluation*, *agreement* and *disagreement*.

3.2 Datasets

We choose three vastly different datasets to evaluate cross-domain transfer.

The **Reddit Dataset (CMV)** consists of about 290,000 debate threads from the subreddit r/ChangeMyView (CMV) (Tan et al., 2016). Each argument is marked as successful, unsuccessful, or neutral, based on whether or not the reply in the thread received a delta from the original poster. The "unsuccessful" label is used when an attempt to convince was made but failed. The “neutral” label refers to texts that are not argumentative, that is, they are not trying to convince the listener of anything. These may be unrelated to the topic at hand or may be simple sentences or phrases that provide no additional information, such as “Yes, I agree”, or “Have you heard of XYZ?” where XYZ has no bearing on the argument. These categories follow Tan et al. (2016). In their work, the authors exclude neutral comments and analyze only argumentative ones; in contrast, we retain neutral comments to train our sequence labeling classifiers and enhance the transferability of our framework, since neutral content is common in realistic scenarios.

The **Supreme Court Oral Arguments Dataset (SCOA)** consists of about 70,000 arguments from various proceedings in the Supreme Court (SCOA) (Danescu-Niculescu-Mizil et al., 2012). Each argument is marked as successful, unsuccessful, or neutral, based on whether or not it was made by the side that won the case. Neutral arguments here cover, in addition, cases where the outcome is not clear.

The **Anthropic Persuasion Dataset (AP)** contains a little less than 4000 claims on a range of general topics, with arguments generated by both humans and LLMs supporting these claims. Each claim has an initial human rating for how much they agree with the initial statement and a human rating for how much they agree after hearing the argument. An argument is considered successful if a person changes their rating from a “disagree” rating (0-4) to an “agree” rating (>4), or if they change their rating by two or more points. Otherwise, it is considered unsuccessful. As this dataset contains only arguments and no neutral examples, it is used solely to evaluate the transferability of the other two models.

These three datasets differ significantly, making cross-domain transfer challenging. The average length of an argument in SCOA is several orders of magnitude larger than that of an argument in CMV, and the lengths of arguments in AP are more concentrated in between the two. These trends can be observed in Fig. 2. Similarly, we observe relatively low token overlap between the CMV and SCOA datasets (Fig. 3). The style and structure of the arguments in the SCOA dataset are also vastly different, as seen in the example in Fig. 1, with more formal speech and more nouns of address.

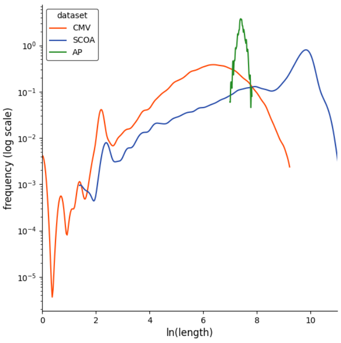


Figure 2: Length Distribution in Datasets

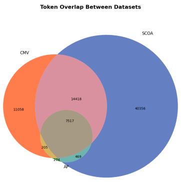


Figure 3: Token Overlap in Datasets

3.3 Model

We propose a simple pipeline that first identifies all argumentation components and their types and then uses this information to predict persuasion.

Identifying Argumentation Components and Their Types. The first step in our approach is to segment the input text and identify the correct argumentation component and semantic type for each segment.

In our implementation, each segment corresponds to a single sentence. Certain conjunctive words tend to indicate the start of a new logical phrase, and so further splitting is done whenever the following are observed: ’but’, ’because’, ’therefore’, ’thus’, ’hence’, ’however’, and ’since’. We break the task into three classification sub-tasks: classifying segments into argumentation components (claims, premises, none), predicting the semantic type of claim (interpretation, rational evaluation, emotional evaluation, agreement, disagreement), and predicting the semantic type of premise (ethos, logos, pathos). All sub-tasks are formulated as sequence-labeling tasks using transformer-based language models.

Predicting Persuasion. Once argumentation components have been identified, we turn our attention to predicting whether an argument is persuasive. Following prior work (Tan et al., 2016; Danescu-Niculescu-Mizil et al., 2012), we define this as a multiclass classification task where the label can be one of: *persuasive*, *not persuasive*,

Model	F1
ArgCompClassifier	0.855
ClaimClassifier	0.696
PremiseClassifier	0.650

Table 1: Text Segment Labeling Classifiers on CMV

Model	SCOA F1	AP F1
ArgCompClassifier	0.702	0.564
SemTypeClassifier	0.368	0.230

Table 2: Text Segment Labeling Classifiers on SCOA and AP (Based on Manual Annotations)

neutral. To use the argumentation taxonomy information, we introduce two special tokens at the beginning of each text segment to identify both the component and its type. We use transformer-based language models for this task.

4 Experimental Results

We describe our experiments using the approach outlined in Sec. 3 to predict persuasion across domains, that is, when training on one dataset and predicting on a different dataset. We also evaluate the performance of our argumentation component and semantic type classifiers. In the CMV data set, this evaluation is based on the corpus of annotated argumentation tags, while in the SCOA and AP datasets, it is based on a much smaller subset of 60 samples that were manually annotated solely for this evaluation.

Experimental Settings. For the argumentation component and semantic type classifiers, we fine-tune DistilBERT (Sanh et al., 2020), while for our cross-domain persuasion classifiers, we fine-tune both DistilBERT (results in Appendix C) and RoBERTa (Liu et al., 2019). Each input is truncated to the maximum length allowed by BERT (512). In all cases, we use 4-fold cross-validation. We test the following combinations:

1. *Segment Labels:* Models were trained using only the argumentation component tags for each segment (ArgComps), using both the argumentation component tags and the semantic type tags (SemTypes), and using neither (Baseline).
2. *Training Data* - Each model is trained on CMV data or SCOA data. Randomly selected subsets of 7500 arguments from the CMV and SCOA datasets are used, with an even split between the number of successful, unsuccessful, and neutral arguments.

The evaluation results for the persuasion classifier model are shown in Table 3. We observe a marked improvement in the model's ability to detect persuasiveness when argument information is explicitly encoded, for both the RoBERTa and DistilBERT classifiers (see Appendix C), showing that this method works irrespective of the base model. However, the performance improvement is minimal on the in-domain task, which could be attributed to segment tag identification errors propagated through the argument component and type classifiers (See Tab. 2). Since argumentation component and semantic type tag annotations are only available for a small subset of the CMV dataset (Hidey et al., 2017), the predictions made by these are treated as ground truth tags while training the final models. While these are able to predict argumentation components and their types quite well for unseen text segments from the CMV set, they were not explicitly trained to classify SCOA or AP segments. Therefore, it is likely that there are more errors in identifying the argumentation components for SCOA, which would also explain why the classifiers trained in this data set offer a lower improvement when using the argumentation information.

Nevertheless, the improvements offered on the cross-domain tasks are encouraging and indicate that this approach has potential. These results show that when a pre-trained transformer encounters types of arguments it has not seen before, such as in a different dataset with different style and vocabulary, the sequence tags do indeed help it identify similarities between training data and the testing data and thus make better predictions. We note that for different datasets, different levels of argumentation information prove useful, with the argumentation component sometimes being sufficient to see a gain, and with the semantic types being required in other cases. Additionally, we observe that the performance of the classifiers with argumentative features on the AP dataset is markedly higher despite the absence of "neutral" text segments, which are present in the training data. This further suggests that our method does indeed generalize.

We have chosen to use plain DistilBERT/RoBERTa, without argumentative features, as our baseline. Our goal is to explore whether incorporating explicit argumentative features in a model improves its performance, which we have shown that it does, rather than obtaining state-of-the-art results for persuasion detection. To accurately gauge whether our hypothesis holds

Model	CMV	SCOA	AP
BaselineLlama	0.375	0.134	0.386
LlamaArgComps	0.389	0.100	0.335
BaselineCMV	**0.570 ± 0.002**	0.311 ± 0.004	0.292 ± 0.003
CMVArgComps	0.570 ± 0.001	**0.377 ± 0.004**	0.319 ± 0.000
CMVSemTypes	0.561 ± 0.000	0.331 ± 0.003	**0.344 ± 0.010**
BaselineSCOA	0.172 ± 0.001	0.689 ± 0.001	0.263 ± 0.001
SCOAArgComps	**0.197 ± 0.002**	**0.698 ± 0.000**	**0.310 ± 0.002**
SCOASemTypes	0.194 ± 0.002	0.686 ± 0.000	0.272 ± 0.003

Table 3: Persuasion Detection Results Across Domains - RoBERTa and Llama3.1

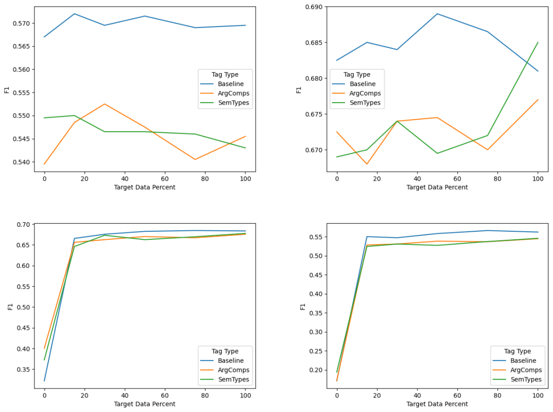


Figure 4: Performance (F1 Score) as more target data is added during training. From left to right: (a) Train on CMV, Eval on CMV (b) Train on SCOA, Eval on SCOA (c) Train on CMV, Eval on SCOA (d) Train on SCOA, Eval on CMV

for other persuasion detection methods, we would potentially have to explore different injection strategies. For completeness, we also include a simple LLM baseline.

LLM Baselines. The transformer-based results are comparable to two-shot LLM baselines: simple prompts passed to a Llama3.1 model, with and without argumentation information. These are visible in Tab. 3.

Training with Target Data. We also performed further experiments to analyze how the performance of these models changed as they saw more training data from the target domain, presented in Figure 4. It is apparent that persuasion detection is a challenging task; all models eventually plateau in performance and are unable to cross a certain threshold.

5 Conclusion and Future Work

While the proposed approach is simple, our findings represent a solid proof of concept that adding inductive bias in the form of argumentative structures and modes of persuasion significantly improves cross-domain persuasion detection, even when this information is noisy. Further, these findings hold for domains with substantial differences in purpose, vocabulary, text length, and style.

In future work, using a more comprehensive scheme to represent arguments, such as Walton's argumentation scheme, could potentially provide a richer representation of the latent structure of the argument. Additionally, other ways of incorporating this information could be explored, such as by cross-attending transformer-based representations of text and graphical networks that model relations between argument components explicitly (Hua et al., 2023), or by combining language model inferences with probabilistic logical inference (Quan et al., 2024; Nafar et al., 2024).

While certain argumentative strategies have been proven to be more effective in certain situations or with certain people (Wang et al., 2019), we have shown the impact of including some underlying indicators that are useful to gauge the persuasiveness of an argument in different domains. These can be further improved by including user- or context-specific information.

Limitations

Our work has two main limitations. Firstly, the scope of our study is small. While we use three datasets from three different domains, these datasets do not cover the full range of domains where persuasion is of importance. A larger study could further verify that our findings hold for other classes of domain variations such as topic and language. Secondly, the proposed approach is somewhat dependent on how well we can identify argument components and their types. We showed that even a noisy representation is beneficial, as we did not fully verify the validity of the intermediate representations for the SCOA case, and still saw an improvement. However, further evaluation is needed to quantify the noise-to-performance ratio.

Regardless of these limitations, we believe that our findings constitute a meaningful, focused contribution that could inform future work in cross-domain persuasion detection.

References

Annye Braca and Pierpaolo Dondio. 2023. Persuasive communication systems: a machine learning approach to predict the effect of linguistic styles and persuasion techniques persuasive communication systems. *Journal of Systems and Information Technology*, 25:1328–7265.

Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. AMPERSAND: Argument mining for PERSuAsive oNline discussions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.

Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. ConvoKit: A toolkit for the analysis of conversations. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 57–60, 1st virtual meeting. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: language effects and power differences in social interaction. In *Proceedings of the 21st International Conference on World Wide Web*, page 699–708.

Brian Rizqi Paradisiaca Darnoto, Daniel Siahaan, and Diana Purwitasari. 2023. Automated detection of persuasive content in electronic news. *Informatics*, 10(4).

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2009–2026, Mexico City, Mexico. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Sebastian Duerr and Peter A. Gloor. 2021. Persuasive natural language generation - a literature review. *ArXiv*, abs/2101.05786.

Subhabrata Dutta, Dipankar Das, and Tanmoy Chakraborty. 2020. Changing views: Persuasion modeling and argument extraction from online discussions. *Information Processing & Management*, 57(2):102085.

James B Freeman. 2011. *Argument Structure:*, 2011 edition. Argumentation Library. Springer, Dordrecht, Netherlands.

Ivan Habernal and Iryna Gurevych. 2016a. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Austin, Texas. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2016b. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

Yilun Hua, Zhaoyuan Deng, and Kathleen McKeown. 2023. Improving long dialogue summarization with semantic graph representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13851–13883, Toronto, Canada. Association for Computational Linguistics.

Rahul Radhakrishnan Iyer and Katia P. Sycara. 2019. An unsupervised domain-independent framework for automated detection of persuasion tactics in text. *ArXiv*, abs/1912.06745.

Chuhao Jin, Kening Ren, Lingzhen Kong, Xiting Wang, Ruihua Song, and Huan Chen. 2024. Persuading across diverse domains: a dataset and persuasion large language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1678–1706, Bangkok, Thailand. Association for Computational Linguistics.

Danial Kamali, Joseph D. Romain, Huiyi Liu, Wei Peng, Jingbo Meng, and Parisa Kordjamshidi. 2024. Using persuasive writing strategies to explain and detect health misinformation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17285–17309, Torino, Italia. ELRA and ICCL.

John Lawrence and Chris Reed. 2015. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 127–136, Denver, CO. Association for Computational Linguistics.

David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Bryan Li, Aleksey Panasyuk, and Chris Callison-Burch. 2024. Uncovering differences in persuasive language in Russian versus English Wikipedia. In *Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia*, pages 21–35, Miami, Florida, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Zhexiong Liu, Mohamed Elaraby, Yang Zhong, and Diane Litman. 2023. Overview of ImageArg-2023: The first shared task in multimodal argument mining. In *Proceedings of the 10th Workshop on Argument Mining*, pages 120–132, Singapore. Association for Computational Linguistics.

Aliakbar Nafar, K. Brent Venable, and Parisa Kordjamshidi. 2024. Teaching probabilistic logical reasoning to transformers. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1615–1632, St. Julian's, Malta. Association for Computational Linguistics.

Kota Shamanth Ramanath Nayak and Leila Kosseim. 2024. Analyzing persuasive strategies in meme texts: A fusion of language models with paraphrase enrichment. *Preprint*, arXiv:2407.01784.

CS Ngai and RG Singh. 2020. Relationship between persuasive metadiscoursal devices in research article abstracts and their attention on social media.

Nikolaos Nikolaidis, Jakub Piskorski, and Nicolas Stefanovitch. 2024. Exploring the usability of persuasion techniques for downstream misinformation-related classification tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6992–7006, Torino, Italia. ELRA and ICCL.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Jakub Piskorski, Nicolas Stefanovitch, Nikolaos Nikolaidis, Giovanni Da San Martino, and Preslav Nakov. 2023. Multilingual multifaceted understanding of online news in terms of genre, framing, and persuasion techniques. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3001–3022, Toronto, Canada. Association for Computational Linguistics.

Xin Quan, Marco Valentino, Louise Dennis, and Andre Freitas. 2024. Enhancing ethical explanations of large language models through iterative symbolic refinement. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–22, St. Julian's, Malta. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Vivian P. Ta, Ryan L. Boyd, Sarah Seraj, Anne Keller, Caroline Griffith, Alexia Loggarakis, and Lael Medema. 2022. An inclusive, real-world investigation of persuasion in language and verbal behavior.

Vivian Ta-Johnson, Ryan Boyd, Sarah Seraj, Anne Keller, Caroline Griffith, Alexia Loggarakis, and Lael Medema. 2022. An inclusive, real-world investigation of persuasion in language and verbal behavior. *Journal of Computational Social Science*, 5.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings* *of the 25th International Conference on World Wide Web*, WWW '16, page 613–624, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Manuel Widmoser, Maria Leonor Pacheco, Jean Honorio, and Dan Goldwasser. 2021. Randomized deep structured prediction for discourse-level processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1174–1184, Online. Association for Computational Linguistics.

A Appendix - LLM Prompts

Llama Prompt Without Argument Information: "I will give you a paragraph of text containing an argument that is trying to be persuasive. Analyze the argument structure and the argumentative strategies employed, and using this, classify the argument as SUCCESSFUL, UNSUCCESSFUL or NEUTRAL. If it a strong argument that is likely to succeed at convincing someone, it is SUCCESSFUL, otherwise it is UNSUCCESSFUL. If it not an argument, say NEUTRAL. Return one of: SUCCESSFUL, UNSUCCESSFUL, NEUTRAL. Some examples to illustrate:
Input: <input1>
Expected Output: <output1>
Input: <input2>
Expected Output: <output2>
Give similar outputs for the argument below:
<text>"

Llama Prompt With Argument Information: "I will give you a paragraph of text containing an argument that is trying to be persuasive. Analyze the argument structure and the argumentative strategies employed, and using this, classify the argument as SUCCESSFUL, UNSUCCESSFUL or NEUTRAL. If it is a strong argument that is likely to succeed at convincing someone, it is SUCCESSFUL, otherwise it is UNSUCCESSFUL. If it not an argument, say NEUTRAL. This paragraph will also have special tags, enclosed in '[]', which states whether the following text segment is a claim, premise, or is neutral, followed by a tag stating the type of claim or premise. Use this information to make your decision. Return one of: SUCCESSFUL, UNSUCCESSFUL, NEUTRAL. Some examples to illustrate: Input: <input1>
Expected Output: <output1>
Input: <input2>
Expected Output: <output2>
Give similar outputs for the argument below:
<text>"

The above prompts are populated with each sample in the test set, and the examples are entered in the input and output spaces. The inputs in the prompt with the argument information also contain argument component and semantic type tags.

B Appendix - Hyperparameters

Information for all hyper-parameters used can be observed in Tab. 4.

C Appendix - DistilBERT Cross-Domain Persuasion Classifier

The results for the persuasion classifier trained using DistilBERT can be found in Tab. 5. There is a slight downturn in performance on the in-domain task, but an improvement in the cross-domain task, as discussed above.

Model	Number of Epochs	Training Batch Size	Optimizer	Learning Rate
ArgCompClassifier	3	8	AdamW	3e-5
ClaimClassifier and PremiseClassifier	5	8	AdamW	3e-5
Baseline Persuasion Classifiers	3	16	AdamW	3e-5
Persuasion Classifiers with Argumentation Info	5	8	AdamW	3e-5

Table 4: DistilBERT and RoBERTa Fine-Tuning Hyperparameters

Model	CMV	SCOA	AP
BaselineCMV	**0.567±0.005**	0.322±0.000	0.314± 0.001
CMVArgComps	0.539±.004	**0.400±.004**	0.316±0.05
CMVSemTypes	0.551±.002	0.375±.002	**0.351±0.003**
BaselineSCOA	0.168±.002	**0.682±.003**	0.192±0.06
SCOAArgComps	0.180±.004	0.672±.000	0.215±0.005
SCOASemTypes	**0.205±.001**	0.669±.001	**0.256±0.002**

Table 5: Persuasion Detection Results Across Domains - DistilBERT

Hallucinated Span Detection with Multi-View Attention Features

Yuya Ogasa[*]
Grad. Sch. of Information Science and Tech.
The University of Osaka
Japan
`ogasa.yuya@ist.osaka-u.ac.jp`

Yuki Arase
School of Computing
Institute of Science Tokyo
Japan
`arase@c.titech.ac.jp`

Abstract

This study addresses the problem of hallucinated span detection in the outputs of large language models. It has received less attention than output-level hallucination detection despite its practical importance. Prior work has shown that attentions often exhibit irregular patterns when hallucinations occur. Motivated by these findings, we extract features from the attention matrix that provide complementary views capturing (a) whether certain tokens are influential or ignored, (b) whether attention is biased toward specific subsets, and (c) whether a token is generated referring to a narrow or broad context, in the generation. These features are input to a Transformer-based classifier to conduct sequential labelling to identify hallucinated spans. Experimental results indicate that the proposed method outperforms strong baselines on hallucinated span detection with longer input contexts, such as data-to-text and summarisation tasks.

1 Introduction

Large Language Models (LLMs) have significantly advanced natural language processing and demonstrated high performance across tasks (Minaee et al., 2024). However, hallucinations persisting in texts generated by LLMs have been identified as a serious issue, which undermines LLM safety (Ji et al., 2024b).

To tackle this challenge, hallucination detection has been actively studied (Huang et al., 2025). Model-level (e.g., (Min et al., 2023)) or response-level (e.g., (Manakul et al., 2023)) hallucination detection has been proposed. However, identification of the hallucinated span is less explored despite its practical importance. Hallucinated span detection enables understanding and manually editing the problematic portion of the output. It also provides clues to mitigate hallucinations in LLM development.

To address this, we tackle hallucinated span detection. While there have been various types of hallucinations (Wang et al., 2024), this study targets hallucinations on contextualised generations that add baseless and contradictive information against the given input context. Motivated by the findings that irregular attention patterns are observed when hallucination occurs (Chuang et al., 2024; Zaranis et al., 2024), we extract features to characterise the distributions of attention weights. Specifically, the proposed method extracts an attention matrix from an LLM by inputting a set of prompt, context, and LLM output of concern. It then assembles features for each token from the attention matrix: average and diversity of incoming attention as well as diversity of outgoing attention, which complementarily capture the attention patterns of language models. The former two features indicate whether attention is distributed in a balanced manner for tokens in the output text. The last feature reveals if an output token was generated by broadly attending to other tokens. These features are then fed to a Transformer encoder with a conditional random field layer on top to conduct sequential labelling to determine whether a token is hallucinated or not.

Experimental results on hallucinated span detection confirmed that the proposed method outperforms strong baselines on data-to-text and summarisation tasks, improving token-level F1 score for 4.9 and 2.9 points, respectively. An in-depth analysis reveals that the proposed method is capable of handling longer input contexts. Our code is available at `https://github.com/Ogamon958/mva_hal_det`.

2 Related Work

This section discusses hallucination detection that utilises various internal states of LLMs.

---

[*] Currently with LY Corporation, Japan. Email: yogasa@lycorp.co.jp

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 381–394
November 8-9, 2025 ©2025 Association for Computational Linguistics

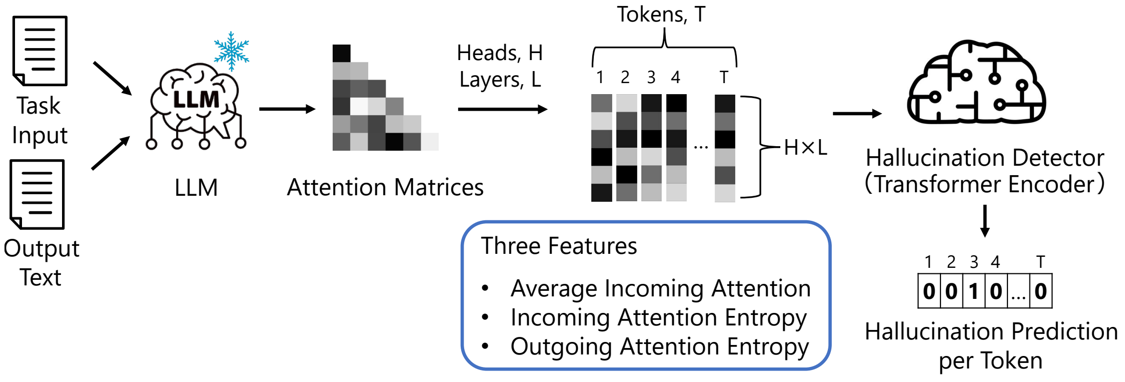

Task Input
Output Text
LLM
LLM
Attention Matrices
Heads, H
Layers, L
Tokens, T
1 2 3 4 T
H×L
Three Features
• Average Incoming Attention
• Incoming Attention Entropy
• Outgoing Attention Entropy
Hallucination Detector (Transformer Encoder)
1 2 3 4 T
0 0 1 0 ... 0
Hallucination Prediction per Token

Figure 1: Overview of the proposed method

Attention-Based Hallucination Detection Lookback Lens (Chuang et al., 2024) is the most relevant method to our study, which identifies hallucinations using only attention matrices. It computes the "Lookback" ratio of attention to assess whether generated tokens attend well to the input context. In contrast, our features primarily focus on the attention of output texts and capture more nuanced and structural attention patterns. ALTI+ (Ferrando et al., 2022; Zaranis et al., 2024) tracks token interactions across layers. ALTI+ has been applied to hallucination detection in machine translation, highlighting cases where the model fails to properly utilise source text information. A drawback of ALTI+ is its computational cost. It computes a token-to-token contribution matrix for each layer and for each attention head. Therefore, memory consumption linearly increases depending on the length of context and output as well as LLM sizes. Indeed, Zaranis et al. (2024) excluded sequences longer than 400 tokens due to GPU memory constraints.

Other Internal States for Hallucination Detection Hallucination detection has also explored various internal states of LLMs other than attention. Xiao and Wang (2021) and Zhang et al. (2023) identify hallucinations as tokens generated with anomalously low confidence based on the probability distribution in the final layer. Azaria and Mitchell (2023) and Ji et al. (2024a) use layer-wise Transformer block outputs to estimate hallucination risk. These studies assume that hallucination detection will be conducted on the same LLM generating output and can access such Transformer block outputs. In contrast, we empirically showed that the proposed method can also be applied to closed LLMs. Further, attention-based methods are distinctive from these studies in that they aim to model inter-token interactions.

3 Proposed Method

The proposed method is illustrated in Figure 1. It conducts sequential labelling, i.e., predicts binary labels that indicate whether a token in text, which has been generated by a certain LLM, is hallucinated or not. Specifically, the proposed method takes a set of prompt, input context, and output generated by an LLM of concern as input to another LLM and obtains the attention matrix of the output text span. It then extracts features from the attention matrix (Sections 3.1 and 3.2). These features are fed to a Transformer encoder model with the prediction head of a conditional random field (CRF) to conduct sequential labelling to identify hallucinated spans (Section 3.3). As the attention matrix provides crucial information for our method, we compare the raw attention and a variation based on the analysis of attention mechanism (Kobayashi et al., 2020) (Section 3.4). We remark that only the hallucination detection model needs training, i.e., the LLM for attention matrix extraction is kept frozen, which makes our method computationally efficient.

Our method applies to both scenarios where the LLM that generated outputs and the LLM for hallucinated span detection are the same or different. In practice, the latter setting is expected to be more common in an era where LLMs are widely used for writing tasks. In addition, we cannot access the internal state of proprietary LLMs. Our experiments assume the scenario where the LLM for generation and the LLM for detection are different.

3.1 Feature Design

Previous studies revealed that irregular patterns of attention are incurred when hallucination oc-

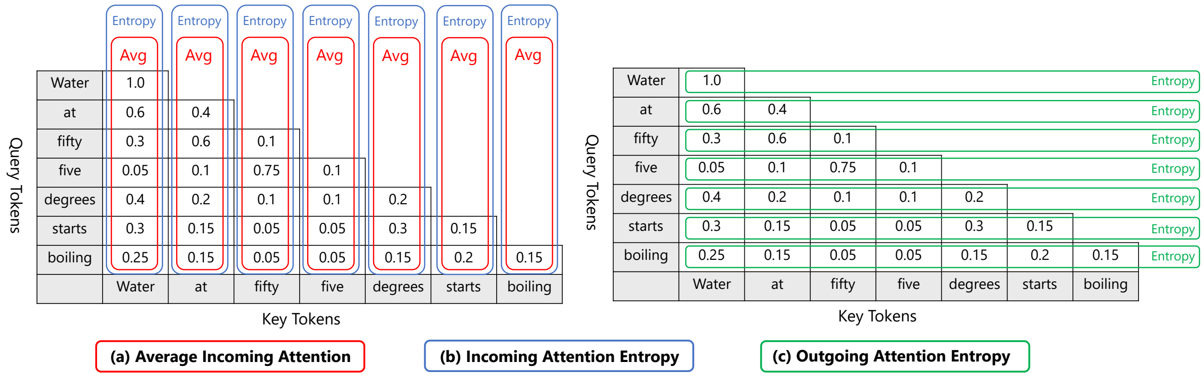

Entropy
Avg
Query Tokens
Water 1.0
at 0.6 0.4
fifty 0.3 0.6 0.1
five 0.05 0.1 0.75 0.1
degrees 0.4 0.2 0.1 0.1 0.2
starts 0.3 0.15 0.05 0.05 0.3 0.15
boiling 0.25 0.15 0.05 0.05 0.15 0.2 0.15
Water at fifty five degrees starts boiling
Key Tokens
(a) Average Incoming Attention
(b) Incoming Attention Entropy
(c) Outgoing Attention Entropy

Figure 2: Feature extraction from attention matrix (these attention values are for illustrative purposes.)

curs (Chuang et al., 2024; Zaranis et al., 2024). Based on these findings, we design features to complementarily capture irregular attentions. Specifically, we extract features providing complementary views of the attention matrix as shown in Figure 2: (a) average attention a token receives (**Average Incoming Attention**), (b) diversity of attention a token receives (**Incoming Attention Entropy**), and (c) diversity of tokens that a token attends to (**Outgoing Attention Entropy**).

Average Incoming Attention We compute the average attention weights that a token receives when generating others. This feature indicates whether certain tokens are influential or ignored in generation. Specifically, it computes the average attention weight in the key direction on the attention matrix as illustrated on the left side of Figure 2.

Incoming Attention Entropy This feature captures the diversity of attention weights, i.e., whether attention is biased toward specific subsets or is more uniformly distributed. It computes the entropy of attention weights in the key direction on the attention matrix as illustrated on the left side of Figure 2.

Outgoing Attention Entropy The final feature models the diversity of tokens that a token attends to when being generated. This indicates whether the model references a narrow or broad range of context for generating the token. Specifically, this feature computes the entropy of attention weights in the query direction on the attention matrix as illustrated on the right side of Figure 2.

Given the complex and diverse nature of attention dynamics, we do not regard individual features as independently effective. Rather, we assume these features *complementary* capture irregular attention patterns due to hallucination by providing views from different angles.

3.2 Feature Extraction

We extract these features for each token from the attention matrix. As notation, the output by an LLM to detect hallucinated span consists of T tokens. The LLM for attention matrix extraction consists of L layers of a Transformer decoder with H heads of multi-head attention.

Average Incoming Attention This feature computes the average attention weights that a token receives when generating other tokens. The attention matrix $\boldsymbol{A}$ is lower triangular due to masked self-attention, meaning each query token i attends only to key tokens j with $1 \leq j \leq i$. Thus, earlier tokens receive attention more often, and tokens close to the end receive attention less often. To compensate for the imbalanced frequency, we adjust the attention weights $\alpha_{i,j}$ as:

$$\alpha'_{ij} = \alpha_{ij} \cdot i. \tag{1}$$

Using the adjusted attention matrix $\boldsymbol{A}'$, the average attention that a key token j receives is computed as:

$$\mu_j^{(\ell,h)} = \frac{1}{T-j+1} \sum_{i=j}^{T} \alpha'^{(\ell,h)}_{ij}, \tag{2}$$

where $1 \leq \ell \leq L$ is the layer index and $1 \leq h \leq H$ is the head index. The final feature vector is obtained by concatenating the average attention weights across all layers and heads:

$$\boldsymbol{v}(j) = \left[\mu_j^{(1,1)}, \mu_j^{(1,2)}, \ldots, \mu_j^{(L,H)}\right] \in \mathbb{R}^{LH} \tag{3}$$

Incoming Attention Entropy To model the diversity of attention a token receives, we use the entropy of the weights. As discussed in the previous paragraph, the attention matrix is lower triangular. To compensate for different numbers of times to receive attention, we normalise an entropy value by dividing by the maximum entropy:

$$\beta_j^{(\ell,h)} = \frac{-\sum_{i=j}^{T} \kappa_{ij}^{(\ell,h)} \log \kappa_{ij}^{(\ell,h)}}{\log(T-j+1)}, \tag{4}$$

$$\kappa_{ij}^{(\ell,h)} = \frac{\alpha_{ij}'^{(\ell,h)}}{\sum_{k=1}^{i} \alpha_{ik}'^{(\ell,h)}}. \tag{5}$$

The final feature vector is a concatenation of the entropy values across layers and heads:

$$\boldsymbol{e}(j) = \left[\beta_j^{(1,1)},\, \beta_j^{(1,2)},\, \ldots,\, \beta_j^{(L,H)}\right] \in \mathbb{R}^{LH} \tag{6}$$

Outgoing Attention Entropy This feature models the diversity of tokens that a token attends to when being generated. Similar to the "Incoming Attention Entropy" feature, we compute the entropy of attention weights of query tokens[1] by dividing by the maximum entropy:

$$\gamma_i^{(\ell,h)} = \frac{-\sum_{j=1}^{i} \alpha_{ij}^{(\ell,h)} \log \alpha_{ij}^{(\ell,h)}}{\log(i)}. \tag{7}$$

The final feature vector is a concatenation of the entropy values across layers and heads:

$$\hat{\boldsymbol{e}}(i) = \left[\gamma_i^{(1,1)},\, \gamma_i^{(1,2)},\, \ldots,\, \gamma_i^{(L,H)}\right] \in \mathbb{R}^{LH} \tag{8}$$

Final Feature Vector The three features $\boldsymbol{v}(j)$ (Average Incoming Attention), $\boldsymbol{e}(j)$ (Incoming Attention Entropy), and $\hat{\boldsymbol{e}}(i)$ (Outgoing Attention Entropy) are concatenated as a final feature vector for hallucination detection. Each feature has LH elements; thus, the final feature vector consists of $3LH$ elements.

3.3 Hallucination Detector

Our hallucination detector consists of a linear layer, a Transformer encoder layer, and a CRF layer on top, as illustrated in Figure 3. To handle *spans*, we employ the CRF layer to model dependencies between adjacent tokens, improving the consistency of hallucinated spans compared to independent token-wise classification.[2] The CRF has been successfully integrated with Transformer-based models for structured NLP tasks (Yan et al., 2019; Wang et al., 2021).

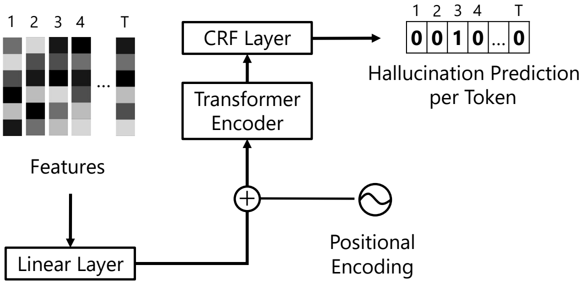


Figure 3: Hallucination Detector

Feature vectors are first standardised to have $zero$ mean and 1 standard deviation per feature type. After standardisation, the feature vector first goes through a linear layer for transformation, which is primarily employed to adapt to various LLMs that can have different numbers of layers and attention heads. Then the transformed vector is input to the transformer layer with positional encoding to incorporate token order information. Finally, the CRF layer predicts a binary label indicating whether a token is hallucinated (label 1) or not (label 0). During inference, the Viterbi algorithm determines the most likely label sequences.

3.4 Attention Weights

Attention weights have been used to analyse context dependency (Clark et al., 2019; Kovaleva et al., 2019; Htut et al., 2019) of Transformer models. Recently, Kobayashi et al. (2020) revealed that the norm of the transformed input vector plays a significant role in the attention mechanism. They reformulated the computation in the Transformer as:

$$\boldsymbol{y}_i = \sum_{j=1}^{T} \alpha_{i,j} f(\boldsymbol{x}_j) \tag{9}$$

where $\alpha_{i,j}$ is the raw attention weight and $f(\boldsymbol{x}_j)$ is the transformed vector of input $\boldsymbol{x}_j$. The transformation function is defined as:

$$f(\boldsymbol{x}) = \left(\boldsymbol{x}\boldsymbol{W}^V + \boldsymbol{b}^V\right)\boldsymbol{W}^O, \tag{10}$$

where $\boldsymbol{W}^V \in \mathbb{R}^{d_{\text{in}} \times d_v}$ and $\boldsymbol{b}^V \in \mathbb{R}^{d_v}$ are the parameters for value transformations and $\boldsymbol{W}^O \in \mathbb{R}^{d_v \times d_{\text{out}}}$ is the output matrix multiplication. Kobayashi et al. (2020) found that frequently occurring tokens often receive high attention weights but have small vector norms, reducing their actual contribution to the output. This suggests that attention mechanisms adjust token influence, prioritising informative tokens over frequent but less meaningful ones.

---

[1] Remind that attention weights are normalised in the query direction.

[2] We empirically confirmed that a linear layer is inferior to CRF in our study.

Dataset	QA	Data2Text	Summarisation
train	$4,584$ $(1,421)$ (31.0%)	$4,848$ $(3,360)$ (69.3%)	$4,308$ $(1,347)$ (31.3%)
valid	450 (143) (31.8%)	450 (315) (70.0%)	450 (135) (30.0%)
test	900 (160) (17.8%)	900 (579) (64.3%)	900 (204) (22.7%)
Total	$5,934$ $(1,724)$ (29.1%)	$6,198$ $(4,254)$ (68.6%)	$5,658$ $(1,686)$ (29.8%)

Table 1: Number of samples in the RAGTruth dataset (Numbers in parentheses indicate the raw number of and percentage of sentences containing at least one hallucination span.)

Hyperparameter	**Search Range**
Learning rate	1e-5 $\sim$ 1e-3
Number of layers	$[2, 4, 6, 8, 10, 12, 14, 16]$
Number of heads	$[4, 8, 16, 32]$
Dropout rate	$0.1 \sim 0.5$
Weight decay	1e-6 $\sim$ 1e-2
Model dimension	$[256, 512, 1024]$
Parameter	**Setting**
Optimizer	AdamW
Batch size	64 (Summrization: 32)
Maximum epochs	150

Table 2: Search ranges of Transformer hyperparameters (upper) and training settings (bottom)

This study compares the effectiveness of raw and the transformed attention weights of Kobayashi et al. (2020). Specifically, we employ the adjusted attention matrix $\boldsymbol{A}_{\text{norm}}$ defined as:

$$\boldsymbol{A}_{\text{norm}} = \boldsymbol{A} \cdot \text{diag}(\|f(\boldsymbol{x})\|), \tag{11}$$

where $\boldsymbol{A}$ is the raw attention weight matrix, and $\text{diag}(\|f(\boldsymbol{x})\|)$ represents a diagonal matrix containing the transformed vector norms.

4 Evaluation

We evaluate the effectiveness of the proposed method for hallucinated span detection.

4.1 Dataset

As the dataset providing hallucination *span* annotation, we employ RAGTruth (Niu et al., 2024)[3], a benchmark dataset that annotates responses generated by LLMs (GPT-3.5-turbo-0613, GPT-4-0613, Llama-2-7B-chat, Llama-2-13B-chat, Llama-2-70B-chat, and Mistral-7B-Instruct). It covers three scenarios of using LLMs in practice, i.e., question answering (QA), data-to-text generation (Data2Text), and news summarisation (Summarisation). RAGTruth provides $18,000$ annotated responses, where hallucinated spans in each response are tagged at the character level. The number of samples is shown in Table 1. As there is no official validation split in RAGTruth, we randomly sampled 450 instances (75 IDs) from the training set for validation.

4.2 Evaluation Metric

The hallucination labels in RAGTruth are provided at the character span level. For example, a hallucination might be annotated with "start: 219, end: 229." We convert these labels into the token level for intuitive interpretation of evaluation results. We employed the same tokeniser of LLM to extract attention matrices.

We compute the token-level precision (Prec) and recall (Rec). Given a set of gold-standard hallucination tokens $\mathcal{Y} = \{y_0, y_1, \cdots, y_N\}$ and predicted hallucination tokens $\hat{\mathcal{Y}} = \{\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_M\}$,

$$\text{precision} = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\hat{\mathcal{Y}}|}, \text{recall} = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\mathcal{Y}|}. \tag{12}$$

Matching of the gold-standard and predicted tokens is computed in the context of output texts. The primary evaluation metric is the F1 score of token-level hallucination predictions, which is the harmonic mean of precision and recall. Following the RAGTruth evaluation scheme, we used the micro-average of precision, recall, and F1.

4.3 Implementation

The proposed method consists of the linear layer, the Transformer encoder layer, and the CRF layer. The settings of the Transformer layer, i.e., the numbers of layers and attention heads, the dimensions, and the dropout rate, were tuned together with other hyperparameters of learning rate and weight decay using the Data2Text task, as it provides the largest samples. We apply the same hyperparameters for

---

[3] `https://github.com/ParticleMedia/RAGTruth`

Methods	LLM	QA			Data2Text			Summarisation		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Ours$_{\text{raw}}$	Llama	47.7	**68.7**	56.3	**55.6**	55.0	**55.3**	51.1	36.7	42.7
Ours$_{\text{norm}}$		57.4	54.0	55.6	53.4	**57.1**	55.2	51.0	**39.5**	**44.5**
Fine-tuning		**62.8**	56.9	**59.7**	55.4	46.2	50.4	**52.0**	34.6	41.6
Lookback Lens		53.5	7.6	13.2	0.0	0.0	0.0	0.0	0.0	0.0

Table 3: Hallucinated span detection results on Llama-3-8B-Instruct. The proposed method is denoted as "Ours" with variations of raw attention ("raw") or the transformed attention ("norm"). It outperformed the baselines on tasks with longer input contexts, i.e., Data2Text and Summarisation.

	QA		Data2Text		Summ.	
	In	Out	In	Out	In	Out
Mean	400	140	788	199	723	136
Max	646	437	1, 499	406	2, 063	412
Min	244	9	517	69	225	16

Table 4: Numbers of tokens of context ('In') and output ('Out') (measured using Llama-3-8B-Instruct tokeniser).

other tasks. In this study, we used the Optuna library[4] to perform hyperparameter search in the ranges shown in the upper rows of Table 2. The setting of the model with the highest F1 score was selected for formal evaluation.

Table 2 bottom shows training settings: we used AdamW (Loshchilov and Hutter, 2019) optimizer with the batch size of 64 (32 for Summarisation). We employed early stopping on training: training was terminated if the F1 score on the validation set did not improve for 10 consecutive epochs. The maximum training epoch was set to 150.

As the LLM to obtain attention matrices, we employed the recent smaller yet strong models of Llama-3-8B-Instruct (Touvron et al., 2023; Llama Team, 2024) and Qwen2.5-7B-Instruct (Team, 2025) (see Appendix A.2 for details). We adapted the template by Niu et al. (2024) for promoting. Notice that these LLMs are different from the ones used to create the RAGTruth dataset, which simulates the scenario where we cannot access the LLMs generated outputs for hallucinated span detection.

4.4 Baselines

We compared the proposed method to two baselines employing the same LLMs as our method.

Fine-tuned LLMs Although straightforward, fine-tuned LLMs serve as a strong baseline (Niu et al., 2024). We fine-tuned the LLMs using the prompt of Niu et al. (2024) with instructions to predict hallucinated spans. More details are provided in Appendix A.3.

Lookback Lens We employed Lookback Lens (Chuang et al., 2024), which also utilises the attention matrix for hallucination detection. It computes the "Lookback" ratio; the ratio of attention weights on the input context versus newly generated tokens. The Lookback feature is input to a logistic regression model to predict the probability of a token being hallucinated.[5] We regarded tokens for which the predicted probabilities are equal to or larger than 0.5 as hallucination, following the traits of the logistic regression classifier. We used the author's implementation[6] for the Lookback Lens model training.

4.5 Experimental Results

The experimental results on Llama-3-8B-Instruct are shown in Table 3. The proposed method is denoted as "Ours" with variations of using raw attention weights (denoted as "raw") and the transformed attention weights (denoted as "norm").

The proposed method outperformed both the fine-tuning and Lookback Lens for hallucinated span detection in Data2Text and summarisation, achieving the highest token-level F1 scores. On QA, the proposed method tends to have higher recall yet lower precision, i.e., it tends to overly detect hallucinations. A possible factor is shorter lengths of input context. Table 4 shows the numbers of tokens in context and output texts. QA has

---

[4] `https://optuna.org/`

[5] Lookback Lens can also conduct span-level prediction by segmenting texts using a sliding window. For direct comparison to our method, we used the token-level variant (i.e., window size is one).

[6] `https://github.com/voidism/Lookback-Lens`

Source text: `[...] From the giant sequoias of Yosemite to the geysers of Yellowstone, the United States' national parks were made for you and me. And for Saturday and Sunday, they're also free. Though most of the National Park Service's 407 sites are free year-round, the 128 parks that charge a fee - like Yellowstone and Yosemite - will be free those two days. It's all part of National Park Week, happening April 18 through April 26, and it's hosted by the National Park Service and the National Park Foundation. [...]`

Output summary: `National Park Service offers free admission to 128 parks, including Yellowstone and Yosemite, on April 18-19 and 25-26, as part of National Park Week.`

Ground Truth: `on April 18-19 and 25-26`
Ours$_{\text{raw}}$: `April 18-19 and 25-26`
Fine-tuning: - (Detection failed)

Table 5: Hallucination detection example (Summarisation)

Methods	QA				Data2Text				Summarisation			
	0–2	2–4	4–6	6–8	0–2	2–4	4–6	6–8	0–2	2–4	4–6	6–8
Ours$_{\text{raw}}$	27.7	–	48.6	59.4	**33.0**	–	**52.6**	**63.3**	0.0	**42.3**	28.5	54.4
Ours$_{\text{norm}}$	25.1	–	41.1	61.0	**33.0**	–	51.2	61.9	0.0	41.9	30.5	**59.0**
Fine-tuning	**38.4**	–	**52.7**	**62.3**	23.8	–	45.8	57.9	0.0	41.0	**31.4**	56.4

Table 6: Token-level F1 scores of hallucinated span detection per different hallucination ratios (Llama-3-8B-Instruct). "–" indicates there was no sample falling in the corresponding bin.

significantly shorter contexts on average compared to Data2Text and summarisation, while the output lengths are similar. This result may imply that the proposed method better handles tasks where consistency with long context is important, like summarisation. We conduct further analysis in Sections 4.6 and 4.7.

For attention weights, the effectiveness of the raw and transformed attention weights depends on tasks. The raw attention weights performed higher in QA, while the transformed weights outperformed the raw attention in summarisation, and they are comparable on Data2Text.

Lookback Lens consistently exhibited the lowest F1 scores.[7] Our inspection confirmed that Lookback Lens overfitted the majority class, i.e., no hallucination. Hallucinated spans are much more infrequent compared to the no-hallucination tokens. This implies that making a binary decision based on the predicted hallucination probability is non-trivial. Furthermore, Lookback Lens seems to have struggled to handle longer input contexts, i.e., Data2Text and summarisation tasks, in contrast to the proposed method. This may be because the Lookback Lens strongly depends on attention weights for the input context. We evaluated the combination of features of Lookback Lens and ours to see if they are complementary. As a result, no improvement was observed; possibly because our "Outgoing Attention Entropy" feature also takes the input context into account.

Table 5 presents an example of hallucination detection on summarisation. In the output text, the red-coloured span indicates the hallucination. While the Fine-tuning failed to detect the hallucination, the proposed method successfully identified the span very close to the ground truth (only missing a preposition). Further examples are in Appendix B.

4.6 Effects of Hallucination Ratio

Intuitively, the ratio of hallucinated tokens in a text affects the performance. When the frequency of hallucinations is small, detection should become more challenging. Table 6 shows the token-level F1 scores on different percentages of hallucinated tokens. These results confirm that the intuition holds true. Across methods and tasks, higher F1 scores were achieved when hallucinated tokens were more frequent.

Another interesting observation is that the effect of task type is dominant than the hallucinated token ratio. Table 6 shows that the superior method is consistent across different frequencies of hallucinated tokens within the same task.

---

[7] This looks largely different from the original paper. We remark that in addition to the experimental dataset difference, the original paper reported AUROC.

Methods	LLM	QA			Data2Text			Summarisation		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Ours$_{\text{raw}}$	Qwen	38.5	**73.7**	50.6	53.5	**57.1**	55.2	49.6	**35.7**	**41.5**
Ours$_{\text{norm}}$		39.0	64.7	48.7	55.5	55.3	**55.4**	49.3	33.6	39.9
Fine-tuning		**60.1**	57.1	**58.6**	**58.9**	51.4	54.9	**62.0**	30.0	40.4
Lookback Lens		46.6	5.6	9.9	50.0	0.0	0.0	0.0	0.0	0.0

Table 7: Hallucinated span detection results on Qwen2.5-7B-Instruct

QA (Total Tokens: 124,817)

Methods	SInfo	EInfo	SConf	EConf	All
Ours$_{\text{raw}}$	**74.1**	**74.4**	–	4.0	**68.7**
Ours$_{\text{norm}}$	50.6	60.0	–	3.8	54.0
Fine-tuning	48.7	63.8	–	**7.8**	56.9
Hal. Tokens	1,020	4,742	–	501	6,263

Data2Text (Total Tokens: 178,343)

Methods	SInfo	EInfo	SConf	EConf	All
Ours$_{\text{raw}}$	29.4	50.5	**7.3**	64.7	55.5
Ours$_{\text{norm}}$	**37.8**	**52.7**	**7.3**	**64.8**	**57.1**
Fine-tuning	35.8	51.6	0.0	43.7	46.2
Hal. Tokens	595	3,118	41	3,580	7,334

Summarisation (Total Tokens: 121,248)

Methods	SInfo	EInfo	SConf	EConf	All
Ours$_{\text{raw}}$	**65.2**	46.5	**8.5**	16.4	36.7
Ours$_{\text{norm}}$	49.7	**51.3**	**8.5**	18.5	**39.5**
Fine-tuning	44.9	43.7	8.1	**18.6**	34.6
Hal. Tokens	187	2,067	71	1,160	3,485

Table 8: Recall of hallucinated span detection per hallucination type (Llama-3-8B-Instruct)

4.7 Effects of Hallucination Type

We further analysed the hallucination detection capability of the proposed method for different hallucination types. RAGTruth categorises hallucinations into four types: Subtle Introduction of Baseless Information (**SInfo**) and Evident Introduction of Baseless Information (**EInfo**) indicate whether the output text subtly adds information or explicitly introduces falsehoods. Subtle Conflict (**SConf**) and Evident Conflict (**EConf**) indicate whether the output alters meaning or directly contradicts the input text. For more details, see Niu et al. (2024).

Table 8 shows detection recalls for different hallucination types.[8] For Data2Text, the recall of Evident Conflict is significantly higher than SInfo and EInfo. This result indicates that the proposed method better captures conflicting information against input context than baseless information introduced by LLMs. The trend is the opposite on QA and summarisation, where the proposed method achieved much higher recall on SInfo and EInfo than on SConf and EConf, which implies that baseless information was easier to capture for the proposed method. These results indicate that detection difficulties of different hallucination types can vary depending on tasks.

4.8 Performance on Qwen

Table 7 shows the results on Qwen2.5-7B-Instruct. While the results are consistent with Table 3, Qwen was consistently inferior to Llama regarding the proposed method, which should be attributed to different implementations of their attention mechanisms. Specifically, Llama-3-8B-Instruct has 32 layers and 32 attention heads, while Qwen2.5-7B-Instruct has 28 layers and 28 heads. Qwen has fewer numbers of layers and attention heads, and thus its feature dimension is smaller than Llama. In addition, the parameters in multi-head attention are more aggressively shared in Qwen. These differences may affect the attention features extracted from Qwen. More details of the differences between Llama and Qwen are discussed in Appendix A.2.

5 Conclusion

We proposed the hallucinated span detection method using features that assemble attention weights from different views. Our experiments confirmed that these features are useful in combination for detecting hallucinated spans, outperforming a previous method that also uses attention weights.

This study focused on hallucination detection, but our method may also apply to broader abnormal behaviour detection of LLMs. As future work, we plan to explore its potential for detecting backdoored LLMs (He et al., 2023), which behave normally on regular inputs but produce malicious outputs when triggered. Since our approach analyses

---

[8] Precision (and thus F1) is difficult to compute because it is non-trivial to decide to which category does detected hallucination belong.

attention distributions, it may detect anomalous attention patterns caused by the triggers.

Limitations

While we confirmed the effectiveness of the proposed method on two models: Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, there are lots more LLMs. The effectiveness of our method when applied to attention mechanisms from other models remains unverified. In addition, our experiments are limited to the English language. We will explore the applicability of our method to other languages by employing multilingual LLMs.

Our method requires training data that annotates hallucinated spans, which is costly to create. A potential future direction is an exploration of an unsupervised learning approach. The success of the current method implies that our features successfully capture irregular attention patterns on hallucination. We plan to train our method only on non-hallucinated human-written text. We then identify hallucinations as instances in which attention patterns deviate from the learned normal patterns.

Acknowledgement

We sincerely thank Professor Tomoyuki Kajiwara for his insightful comments and valuable discussions that greatly improved this work. This work was supported by JST K Program Grant Number JPMJKP24C3, Japan.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4895–4901.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It‘s Lying. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 967–976.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1419–1436.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT‘s Attention. In *Proceedings of the Workshop on Analysing and Interpreting Neural Networks for NLP (BlackboxNLP)*, pages 276–286.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards Opening the Black Box of Neural Machine Translation: Source and Target Interpretations of the Transformer. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8756–8769.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. Mitigating Backdoor Poisoning Attacks through the Lens of Spurious Correlation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 953–967.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do Attention Heads in BERT Track Syntactic Dependencies? *arXiv:1911.12246*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024a. LLM Internal States Reveal Hallucination Risk Faced With a Query. In *Proceedings of the Workshop on Analysing and Interpreting Neural Networks for NLP (BlackboxNLP)*, pages 88–104.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2024b. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374.

AI @ Meta Llama Team. 2024. The Llama 3 Herd of Models. *arXiv:2407.21783*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. *arXiv:1711.05101*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9004–9017.

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

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large Language Models: A Survey. *arXiv:2402.06196*.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10862–10878.

Qwen Team. 2025. Qwen2.5 technical report. *arXin:2412.15115*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*.

Chenyi Wang, Tianshu Liu, and Tiejun Zhao. 2021. HITMI&T at SemEval-2021 Task 5: Integrating Transformer and CRF for Toxic Spans Detection. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 870–874.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024. Do-Not-Answer: Evaluating Safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL*, pages 896–911.

Yijun Xiao and William Yang Wang. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 2734–2744.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv:1911.04474*.

Emmanouil Zaranis, Nuno M Guerreiro, and Andre Martins. 2024. Analyzing Context Contributions in LLM-based Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 14899–14924.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing Uncertainty-Based Hallucination Detection with Stronger Focus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 915–932.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 400–410.

A Details of Experiment Settings

A.1 Computational Environment

All the experiments were conducted on NVIDIA RTX A6000 (48GB memory) GPUs. For training the Transformer encoder of the proposed method, we used 2 GPUs. For fine-tuning the LLM, we used 4 GPUs in parallel.

A.2 LLM Details

Llama-3-8B-Instruct has 32 layers and 32 attention heads, while Qwen2.5-7B-Instruct has 28 layers and 28 heads. Both models replace standard Multi-Head Attention (MHA) with Grouped-Query Attention (GQA) (Ainslie et al., 2023), but Llama-3 uses more layers and heads than Qwen2.5.

MHA assigns each query to a single key-value pair, whereas GQA allows multiple queries to share a key-value pair, reducing the number of trainable parameters. Llama-3-8B-Instruct processes 32 queries while reducing the number of keys and values to 8, so each key-value pair corresponds to 4 queries. In contrast, Qwen2.5-7B-Instruct processes 28 queries and reduces the number of keys and values to 4, making each key-value pair correspond to 7 queries.

We conjecture these differences were reflected in the different performances of Llama and Qwen in our method.

A.3 Fine-Tuning

Fine-tuning was conducted using LLaMA-Factory (Zheng et al., 2024)[9], a library specialized for fine-tuning LLMs. The fine-tuning parameters are shown in Table 9. The fine-tuned model predicts the hallucinated span by predicting character indexes. If a hallucination label changes within a single token in predictions, the entire token is considered as being hallucinated.

---

[9] `https://github.com/hiyouga/LLaMA-Factory`

Parameter	**Value**
Fine-tuning method	full fine-tuning
Learning rate	5e-6
Batch size	1
Number of epochs	3
Optimizer	AdamW
Warmup steps	10

Table 9: Fine-tuning Parameters

A.4 Prompts of RAGTruth

The prompts used in our experiments are shown in Table 10 and Table 11.

B Hallucination Detection Examples

Table 12 presents hallucination detection results in the QA task. The Fine-tuning baseline incorrectly judged the non-hallucinated span as hallucinated and largely overlooked the truly hallucinated span. In contrast, the proposed method mostly correctly identified the hallucinated span.

Table 13 presents hallucination detection results in the summarisation task where the proposed method failed. In the first example, the proposed method overlooked the hallucinated span. In the second example, the proposed method mistook the non-hallucinated span as hallucinated.

QA Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Briefly answer the following question:
{question}
Bear in mind that your response should be strictly based on the following three passages:
{passages}
In case the passages do not contain the necessary information to answer the question, please
reply with:
"Unable to answer based on given passages."
output:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{answer} <|eot_id|>
```

Data2Text Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Instruction:
Write an objective overview about the following local business based only on the provided
structured data in the JSON format.
You should include details and cover the information mentioned in the customers’ review.
The overview should be 100 - 200 words. Don’t make up information.
Structured data:
{json_data}
Overview:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{Converted text} <|eot_id|>
```

Summarisation Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Summarize the following news within {word count of the summary} words:
{text to summarize}
output:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{summary} <|eot_id|>
```

Table 10: Prompts for RAGTruth (Using Llama-3-8B-Instruct)

QA Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the instructions.<|im_end|>
<|im_start|>user
Briefly answer the following question:
{question}
Bear in mind that your response should be strictly based on the following three passages:
{passages}
In case the passages do not contain the necessary information to answer the question, please
reply with:
"Unable to answer based on given passages."
output:<|im_end|>
<|im_start|>assistant
{answer}<|im_end|>
```

Data2Text Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the instructions.<|im_end|>
<|im_start|>user
Instruction:
Write an objective overview about the following local business based only on the provided
structured data in the JSON format.
You should include details and cover the information mentioned in the customers' review.
The overview should be 100 - 200 words. Don't make up information.
Structured data:
{json_data}
Overview:<|im_end|>
<|im_start|>assistant
{Converted text}<|im_end|>
```

Summarisation Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the instructions.<|im_end|>
<|im_start|>user
Summarize the following news within {word count of the summary} words:
{text to summarize}
output:<|im_end|>
<|im_start|>assistant
{summary}<|im_end|>
```

Table 11: Prompts for RAGTruth (Using Qwen2.5-7B-Instruct)

Key Part of the Question

Briefly answer the following question: differences between elements, compounds, and mixtures. Bear in mind that your response should be strictly based on the following three passages: ... (Following this, three passages appear, but there is no information on the bonding of elements or compounds. The content primarily focuses on mixtures.)

Ground Truth Label

... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components.

Ours$_{\text{raw}}$

... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components.

Fine-tuning

... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components. (Incorrect detection in point 3 + incomplete detection in point 4. The discussion on separation in point 3 is fully described in the original text.)

Table 12: Hallucination detection example (QA)

Example 1

Key Part of the Target Sentence for Summarisation

... Doug Ducey signed legislation to allow Arizonans to get any lab test without a doctor's order. Freedom of information - always sounds like a good thing. ... (The target sentence for summarisation contains no mention of Doug Ducey being the governor of Texas. In fact, he was a former governor of Arizona, making this incorrect.)

Ground Truth Label

The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ...

Ours$_{\text{raw}}$

The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ... (Detection failed)

Fine-tuning

The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ...

Example 2

Key Part of the Target Sentence for summarisation

... Still, the average monthly benefit for retired workers rising by $59 to $1,907 will undoubtedly help retirees with lower and middle incomes to better cope with inflation. ... ($1907-$59=$1848 increase)

Ground Truth Label

... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ...

Ours$_{\text{raw}}$

... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ... (False detection)

Fine-tuning

... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ...

Table 13: Hallucination detection example (Summarisation)

AdvERSEM: Adversarial Robustness Testing and Training of LLM-based Groundedness Evaluators via Semantic Structure Manipulation

Kaustubh D. Dhole, Ramraj Chandradevan, Eugene Agichtein
Department of Computer Science
Emory University
`{kaustubh.dhole, eugene.agichtein}@emory.edu`

Abstract

Evaluating outputs from large language models (LLMs) presents significant challenges, especially as hallucinations and adversarial manipulations are often difficult to detect. Existing evaluation methods lack robustness against subtle yet intentional linguistic alterations, necessitating novel techniques for reliably assessing model-generated content. Training accurate and robust groundedness evaluators is key for mitigating hallucinations and ensuring the alignment of model or human-generated claims to real-world evidence. However, as we show, many models, while optimizing for accuracy, lack robustness to subtle variations of claims, making them unsuitable and brittle in real-world settings where adversaries employ purposeful and deceitful tactics like hedging to deceive readers, which go beyond surface-level variations. To address this problem, we propose **AdvERSEM**, a controllable adversarial approach to manipulating LLM output via Abstract Meaning Representations (AMR) to generate attack claims of multiple fine-grained types, followed by automatic verification of the correct label. By systematically manipulating a unique linguistic facet **AdvERSEM** provides an interpretable testbed for gauging robustness as well as useful training data. We demonstrate that utilizing these AMR manipulations during training across multiple fact verification datasets helps improve the accuracy and robustness of groundedness evaluation while also minimizing the requirement of costly annotated data. To encourage further systematic evaluation, we release **AdvERSEM-Test**, a manually verified groundedness test-bed.[1]

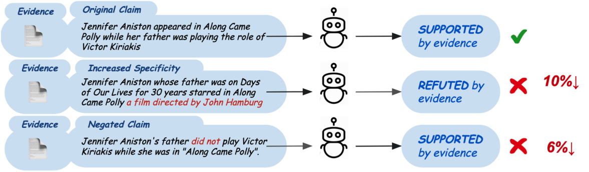

Evidence
Original Claim
Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis
SUPPORTED by evidence
✓
Evidence
Increased Specificity
Jennifer Aniston whose father was on Days of Our Lives for 30 years starred in Along Came Polly a film directed by John Hamburg
REFUTED by evidence
✗
10%↓
Evidence
Negated Claim
Jennifer Aniston's father did not play Victor Kiriakis while she was in "Along Came Polly".
SUPPORTED by evidence
✗
6%↓

Figure 1: Groundedness Evaluators falter when the claim is made more specific or is negated

1 Introduction

Evaluating the reliability of human or model-generated claims typically involves human judgment, which can be costly, and insufficiently sensitive to subtle manipulations in generated text. Automatic evaluators like LLM-Judges offer scalable alternatives for assessing critical properties like groundedness, and factual accuracy, yet these evaluators themselves often rely on LLMs and thus inherit their limitations, including susceptibility to hallucinations, adversarial perturbations and the transformers' non-compositional nature of training (Nandi et al., 2025). Developing robust automatic evaluators is particularly challenging, as subtle semantic changes (Lee et al., 2025; Raina et al., 2024), purposeful negations, or intentional manipulations like hedging (Paige et al., 2024) can cause dramatic degradation in evaluation reliability. Besides, most evaluation testbeds rarely offer fine-grained performance assessments, providing almost zero feedback to evaluation assessors. Hence, to improve trustworthiness, it is vital that groundedness are robust as well as provide fine-grained feedback of performance.

Groundedness evaluation or fact verification refers to the alignment of human or model-generated claims with real-world evidence. This alignment is crucial for assessing the truthfulness of statements against established knowledge. However, current language models often exhibit vulnerabilities that compromise their groundedness. For instance, the Llama-3.1-7B model fails to consistently predict the factual correctness of claims when claims are slightly altered to make them more specific or when they are simply negated as shown in Figure 1. In this work, we focus on improving groundedness evaluation by fine-tuning on adversarially generated examples. Specifically, using popular fact verification datasets, we construct ad-

---

[1] `https://github.com/emory-irlab/adversem`

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 395–408
November 8-9, 2025 ©2025 Association for Computational Linguistics

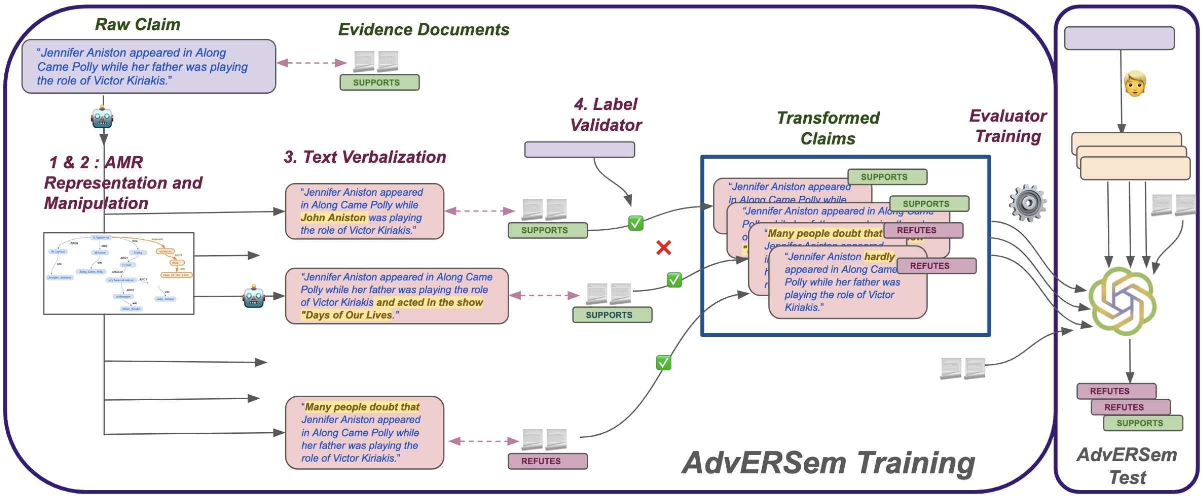

Raw Claim
"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
Evidence Documents
SUPPORTS
1 & 2 : AMR Representation and Manipulation
3. Text Verbalization
"Jennifer Aniston appeared in Along Came Polly while John Aniston was playing the role of Victor Kiriakis."
"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis and acted in the show "Days of Our Lives."
"Many people doubt that Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
SUPPORTS
SUPPORTS
REFUTES
4. Label Validator
Transformed Claims
"Many people doubt that
"Jennifer Aniston hardly appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
SUPPORTS
SUPPORTS
REFUTES
REFUTES
Evaluator Training
AdvERSem Training
REFUTES
REFUTES
SUPPORTS
AdvERSem Test

Figure 2: **AdvERSEM** training and evaluation on **AdvERSEM-Test**.

versarial claims designed to expose strategic vulnerabilities in LLMs. However, generating effective adversarial attacks remains an open research question, requiring careful strategies to ensure examples both challenging and representative examples.

Attempts to create adversarial examples for claim verification have traditionally focused on surface-level perturbations to challenge model robustness (Thorne et al., 2018b). These methods typically involve introducing noise, substituting entities, or making minor lexical alterations to input texts. While such approaches have been instrumental in exploiting LLM vulnerabilities, they often fail to capture actual errors involving deeper semantic and syntactic complexities (Morris et al., 2020) or carefully crafted manipulation like hedging often used to dodge potential disagreement (Hyland, 1998; White, 2003) – for instance journalists may use phrases like "some suggest," "it might appear," or "many doubt" to propose claims while distancing themselves from responsibility or certainty.

Besides, many of these traditional techniques, while revealing some issues related to logical reasoning or factual consistency, rarely provide clarity on specific patterns that evaluators fail to understand. Moreover, these surface-level perturbations can sometimes lead to unnatural or ungrammatical sentences, limiting their effectiveness in real-world applications. To improve groundedness evaluation, it is hence crucial to identify more abstract, structured, and compositional patterns that mimic human language patterns as well as provide fine-grained assessments.

In this paper, we investigate two research questions – ***RQ1***: How can we systematically generate adversarial claims to attack automated groundedness evaluators, and which attacks are particularly effective against SOTA LLM-based evaluators? ***RQ2***: How can we generate useful adversarial data for training groundedness evaluators that are robust to such attacks?

To investigate these research questions, we propose a novel framework that systematically generates adversarial examples through manipulations in a higher-order logical space, particularly through Abstract Meaning Representations (AMR). By extracting away from syntactic variations, AMR provides a structured, graph-based representation of sentence semantics. By manipulating claims in their abstract representations, we can create challenging test cases that expose specific weaknesses in groundedness evaluators, which provide systematic feedback of groundedness performance while addressing the typical issues. In summary, our contributions are the following:

1. We identify a family of semantics-based adversarial claim manipulations that resemble human-like manipulations like hedging, and show that these manipulations can successfully mislead SOTA LLM-based groundedness evaluators.
2. We propose **AdvERSEM** – a framework for **Adv**ersarial **E**valuation and **R**obustness through **Sem**antic **M**anipulations – an effective, and robust framework to evaluate and train LLMs by assessing their groundedness systematically across multiple fine-grained dimensions.
3. We create a manually-curated holistic adversarial test generated from **AdvERSEM** called **AdvERSEM-Test**, with various types of manipulations that mimick real-world groundedness errors.

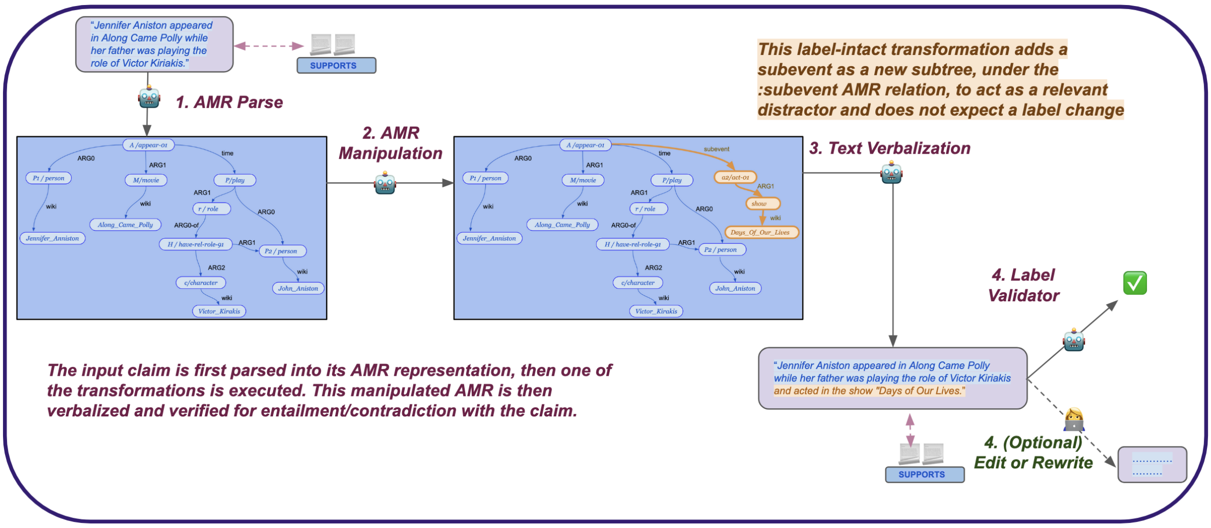

"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
SUPPORTS
1. AMR Parse
2. AMR Manipulation
This label-intact transformation adds a subevent as a new subtree, under the :subevent AMR relation, to act as a relevant distractor and does not expect a label change
3. Text Verbalization
4. Label Validator
The input claim is first parsed into its AMR representation, then one of the transformations is executed. This manipulated AMR is then verbalized and verified for entailment/contradiction with the claim.
"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis and acted in the show "Days of Our Lives."
SUPPORTS
4. (Optional) Edit or Rewrite

Figure 3: Illustration of Adversarial Manipulation using Abstract Meaning Representation (AMR). In this example Specificity of a claim is manimuplated by adding a ":subevent" subgraph.

4. We demonstrate how **AdvERSEM** can be used to identify potential vulnerabilities and be used to improve accuracy and robustness on multiple fact verification benchmarks.

2 Related Work

We now discuss related research to place our contributions in context.

False or adversarial claims designed to mislead human readers, and now machine-reading models, have been long studied and analyzed. Apart from deliberate biases exhibited by journalists and traditional information providers, many frequently employ rhetorical techniques such as hedging to preempt disagreement (Hyland, 1998; White, 2003; Raina et al., 2024) or careful wordplay aimed at misleading or unduly influencing readers. Recently, the problem of misinformation and disinformation has become critically important due to their unprecedented speed and reach, further amplified by digital platforms and recent LLMs. Specifically, Vosoughi et al. (2018) demonstrated that false news spreads significantly faster, farther, and more broadly than truthful information, underscoring the urgent need for improved claim verification methods. Recent studies, such as those by Zhou et al. (2023), have emphasized this growing challenge by illustrating how AI-generated misinformation convincingly integrates fabricated details with truthful elements, effectively evading traditional detection approaches.

Hence, automated evaluation of claims, or fact verification, has been an active area of research. Works such as MultiFC (Augenstein et al., 2019), LIAR (Wang, 2017), and AVeriTeC (Schlichtkrull et al., 2023), emphasize real-world claims verified by journalists or professionals, offering more diverse and context-rich challenges. FEVEROUS (Aly et al., 2021) expands on earlier work by incorporating structured data like tables, while domain-specific datasets like SciFact (Wadden et al., 2022) and COVID-Fact (Saakyan et al., 2021) target scientific or health misinformation.

To improve the robustness of fact verification systems, prior works have produced a range of datasets used to train and evaluate fact-checking systems. Datasets like FEVER (Thorne et al., 2018a), FEVER 2.0 (Thorne et al., 2018b), and VitaminC (Schuster et al., 2021) focus on generating adversarial or subtly false claims to test model robustness, often using Wikipedia as a source, but most of their adversarial generations either use uncontrolled manipulations or use flat first-order logic representations like OpenIE triples (Alonso-Reina et al., 2019). ConQRet (Dhole and Agichtein, 2024; Dhole et al., 2025; Dhole, 2025) provides a benchmark of long-form generations with controlled hallucinations for fine-grained evaluation of groundedness in retrieval-augmented systems.

While previous work has considered adversarial claim generation to evaluate and train fact checking models, our work is the first, to our knowledge, to systematically investigate the specific types of semantic manipulation that can successfully attack state-of-the-art LLM-based fact checkers, and propose a controllable way to generate successful semantic manipulation attacks that can be used to both evaluate existing fact checkers, and to automatically augment training data to make the fact checkers more robust to a range of attacks.

3 Proposed Method: AdvERSEM

We now describe our proposed method, **AdvERSEM** starting with the end-to-end overview of the method and then diving into the details of the implementation for each module.

3.1 Training Fact Verifiers: Overview

Given a model-generated or a human-written claim, **c**, and evidence text **D**, the task of groundedness evaluation refers to identifying the veracity **v** of the claim, i.e., whether the given evidence supports or refutes the claim, where $\mathbf{v} \in \{\text{SUPPORTS}, \text{REFUTES}\}$.

AdvERSEM (i) parses a claim into its AMR graph, (ii) applies multiple manipulations, (iii) verbalises the edited graphs, and (iv) trains (and evaluates) a claim verifier on the mixed i.e. original + adversarial pairs—see the full pipeline in Figure 2. A sample **AdvERSEM** manipulation (Subevent Addition) is shown in Figure 3.

3.2 Adversarial Manipulation of Claims using Abstract Meaning Representation

We now delve into the details of adversarial generation of claims to augment the testing and training data for evaluator models, namely steps 1 and 2 in Figure 2.

Our approach is to design controllable adversarial attacks based on Abstract Meaning Representations (AMR) (Banarescu et al., 2013). AMR is a semantic representation that represents sentences as rooted, directed, acyclic graphs, abstracting away from syntactic variations. This abstraction allows for systematic manipulations of sentence meaning through graph transformations. We developed 17 specialized AMR-based manipulations aimed at altering claims in various ways. Each manipulation targets a distinct aspect, as detailed in Table 1. The corresponding prompts used to implement these manipulations are provided in Appendix A. For instance,"*Focus Shift*" manipulation changes the entity being focused upon, while "*Hedging*" attempts to introduce mild uncertainty – i.e. either making the claim appear milder or stronger. "*Polarity Negation*" either adds or removes polarity to flip the veracity, while "*SubEvent Addition*" adds a subevent related to the main event.

We manually validate the different verbalisations of each one of the 17 different AMR manipulations, and finalize 10 of those for which veracity is guaranteed – namely *Topic Addition, Source Addition, Subevent Addition, Structural Reversal, Hedging, Focus Shifting, Scalar Adverb Negation, Polarity Negation (either removal or addition), and Scalar Negation*. Details of the remaining ones are provided in Appendix A and will be used in the future for noisy data augmentation.

We now describe **AdvERSEM** in detail. To create an adversarial test set for each of the defined manipulation types, we employed `GPT4-o` (Achiam et al., 2023) to execute each of the following steps:
1. ***AMR Parsing***: We first convert the input claim c into its AMR graph representation G_c using a 10-shot prompt based on the AMR 1.2.6 specification. We employ the few-shot setting owing to its superior performance and conformity to structure in previous settings (Ettinger et al., 2023). This transformation can be denoted as:

$$G_c = \texttt{AMRParse}(c)$$

where $\texttt{AMRParse}(\cdot)$ is the function that maps natural language claims to their semantic graph representations. An example of the parsing prompt is shown in Figure 4.
2. ***AMR Manipulation***: The AMR graph G_c is then modified according to a specified manipulation action a, guided by a natural language instruction, resulting in a new, manipulated AMR $\tilde{G}_c$:

$$\tilde{G}_c = \texttt{Manipulate}(G_c, a)$$

where $\texttt{Manipulate}(\cdot)$ alters the structure while maintaining plausibility and grammaticality. An example of this process is illustrated in Figure 5.
3. ***Text Verbalization***: Finally, the manipulated AMR $\tilde{G}_c$ is verbalized into a new natural language claim $\tilde{c}$ using a 10-shot prompt:

$$\tilde{c} = \texttt{Verbalize}(\tilde{G}_c)$$

where $\texttt{Verbalize}(\cdot)$ is the AMR-to-text generation function.
For both parsing and verbalization, we format the prompts with examples adapted from the official AMR specification.[2] Examples for each transformation and a sample subevent transformation are shown in Table 3 and in Figure 3, respectively.

---

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

Manipulation	Description	Intact	Flipped
Manipulations (With Veracity Guarantee)			
Specificity: Topic Addition	Attaches background themes or topics to situate the event with (e.g., topic: "movies")	✓	
Specificity: Source Addition	Provides spatial or origin-based context for the event (e.g., source: "scientists")	✓	
Specificity: Subevent Addition	Embeds smaller, related events within the main event to deepen the narrative (e.g., subevent: "local protest")	✓	
Structure Reversal	Re-attaches entities in new relationships by reversing argument structure (e.g., ARG0-of: "movement leader")	✓	
Hedging	Introduces uncertainty (mild or strong), e.g., via modal verbs or 'doubt-01' (e.g., "might")	✓	✓
Focus Shift	Changes the focus of the claim (e.g., shifting the subject) (e.g., from "spokesperson" to "organization")	✓	
Scalar Adverb Negation	Reverses claim polarity by adding or removing scalar adverbs broadly (if positive, weaken; if negative, restore)		✓
Polarity Negation (Addition)	Flips the claim's veracity by adding a negation marker if absent (e.g., polarity neg add: "did not occur")		✓
Polarity Negation (Remove)	Reverses negation by removing an existing negation marker (e.g., polarity neg remove: remove "not")		✓
Specific Scalar Negation	Inserts a scalar negation only when there's no negation (e.g., "barely noticeable")		✓
Entity Substitution	Replaces named entities with aliases or alternatives from the same category to preserve or invert the original claim.	✓	✓
Temp/Numeric Attribute Substitution	Modifies time-based or numerical expres- sions to retain or change the truth value of the sen- tence.	✓	✓

Table 1: Adversarial AMR manipulations. A checkmark indicates the veracity setting(s) in which the manipulation applies. The extended list is shown in Appendix Table 7.

3.3 Sentence Based Manipulation

In the same spirit of AMR-based manipulations, we introduced two adversarial manipulations that operate directly over the claim sentences.

Entity Attribute Substitution Replaces named entities with aliases or alternatives from the same category to preserve or flip veracity.

Temporal and Numerical Attribute Substitution Modifies time-based or numerical expressions to retain or flip the veracity.

3.4 Generated Example Validation

To ensure that our generated examples reflect the intended veracity labels, we apply an entailment-based filter that verifies logical consistency between original and transformed claim pairs $(c, \tilde{c})$. Rather than comparing claims against extensive evidence, we use GPT-4o to evaluate entailment directly between concise claim statements.

We provide GPT-4o with two instructions (see Appendix Figure 6): for ***label-intact*** transformations, the original and transformed claims must entail each other; for ***label-flip*** transformations, they must contradict. Only claim pairs that pass this validation are retained. The resulting dataset is then used to train a more robust evaluator model.

The percentage applicability of each transformation is provided in Appendix Table 8.

3.5 Label Agreement

We also measure the agreement between the final annotations and human annotations (Table 2). We find that the agreement rate in the case of label intact is high but poor for the label flipping case, motivating us to create a manually modified evaluation set that we describe in the upcoming section.

	Label Intact	Label Flip
# Agreed Labels	23	13
Total Measured	24	33
Percentage	.96	.40

Table 2: Label agreement for adversarial claims between human annotations and GPT-4o.

4 Experiments

We now present the choice of datasets and models used along with the corresponding experiments.

4.1 Datasets

We evaluate our performance on 3 datasets using 2 metrics. We consider 1) FEVER 2.0 (Thorne et al., 2018b), which provides more realistic adversarial claims, 2) AVERITEC (Schlichtkrull et al., 2023), whose claims necessitate verification using publicly available noisy sources over the web, and 3) introduce **AdvERSEM-Test** our novel manually verified evaluation set. We measure – the accuracy i.e. performance on the raw dataset, and robustness i.e. accuracy on the **AdvERSEM** transformed sets.

The FEVER 2.0 dataset comprises adversarial examples generated by competing systems in the associated shared task and subsequently refined manually by the task organizers. These examples primarily leverage Wikipedia as their source of evidence. For a more realistic evaluation of large language models, particularly Retrieval-Augmented Generation (RAG) systems, we additionally use the AVERITEC fact verification dataset. AVERITEC includes claims supported by links to publicly accessible websites. We augment AVERITEC by scraping these websites using custom Python scripts and incorporate them as evidence.

Finally, we apply the **AdvERSEM** transformations described in §3 to both the training and test sets of FEVER 2.0 and AVERITEC.

We retain only the claims labeled as either *Supported* or *Refuted* in both datasets. From the FEVER dataset, we select 800 examples, reserving 80 for testing and development. For the AVERITEC dataset, we reserve 120 examples for testing and development out of a total of 1,565.

4.2 AdvERSEM-Test

To create **AdvERSEM-Test**, we pass the first 20 examples of the FEVER 2.0 test set, through each of the **AdvERSEM** transformations. The generated claims are then manually refined to ensure that the transformation's particular change is reflected. Publicly available LLM based interfaces are used as an intermediary if needed. Through this process, we gather 200 adversarial and manually verified claims for systematic fine-grained analysis.[3]

4.3 Training Details

We train our groundedness evaluator using GPT-4o-mini (Achiam et al., 2023) and LLaMA-3.1-8B (Dubey et al., 2024), balancing performance and cost. Models are trained for 3 epochs with a batch size of 32 using a supervised chat completion objective. LLaMA-3.1-8B is trained via LLaMAFactory (Zheng et al., 2024) and HuggingFace (Wolf et al., 2020).

Baselines: We used various zero-shot LLMs, including GPT4o, GPT4o-mini, gemini-2.0-flash, and llama-3.1-8B, by leveraging the prompt as shown in Figure 8.

Regular Trained: We trained our groundedness evaluators on the given human-labeled training set (without any adversarial manipulations) *i.e.* $(\mathbf{D}, \mathbf{c}, \mathbf{v})$ tuples.

AdvERSEM Trained: We additionally include **AdvERSEM**-generated adversarial claims in our training set *i.e.,* $(\mathbf{D}, \mathbf{c}, \mathbf{v}) + (\mathbf{D}, \tilde{\mathbf{c}}, \mathbf{v})$ tuples. We also experiment by choosing the number of training examples from the adversarial set in proportion to the errors (EP) on the FEVER 2.0 dev set. Let M be the set of all manipulation types, and E_m the number of errors for type m. Then the probability of selecting an adversarial example of type m is

$$P(m) = \frac{\mathrm{E}_m}{\sum_{k \in M} \mathrm{E}_k}.$$

5 Results

We now present the results for all our evaluations.

5.1 Accuracy and Robustness

The summarised results are present in Table 4 both on the transformed sets (**T**) as well as the raw (**R**) sets of all the fact verification datasets.

We find that **AdvERSEM**-trained models are significantly more robust than regularly trained models, both in terms of macro as well as micro average across all the 3 benchmarks. **For instance, by including AdvERSEM generated examples, the micro-average performance improves by 7.1% in the case of GPT4o-mini, and 3% in the case of Llama-3.1 over the AdvERSEM-Test.**

Additionally, **AdvERSEM** also keeps the performance on other raw datasets like FEVER 2.0 intact. For instance, GPT4o-mini improves the performance on the FEVER 2.0 test set by 1.2% while significantly improving robustness.

5.2 Fine-Grained Robustness Analysis

Table 5 illustrates the fine-grained performance across various adversarial manipulations of **AdvERSEM-Test**. Models trained using the **AdvERSEM** approach consistently outperform both baseline and regularly trained models, demonstrating substantial gains in robustness across nearly all categories of adversarial manipulation. Notably, **GPT4o-mini trained with AdvERSEM achieves the highest macro-averaged robustness of 85%, significantly surpassing its regular training variant by 6.6% and the zero-shot GPT-4o baseline by approximately 9.6%.** Among label-flipped manipulations, the improvement is particularly marked, with the performance on manipulations such as *scalar negation* and *hedging* improving by approximately 25% and 20%, respectively.

When we look at specific manipulations, we observe that **LLMs fail extensively on label-flipped manipulations, specifically on negations, and perform the poorest on hedging**. Besides, random training data is insufficient to mitigate those errors and in fact may also hurt performance. For instance, when evaluated on scalar negatives (*Specific Scalar Negation*), GPT4o-mini reduced performance from 70% to 55% after being trained on FEVER 2.0.

On the other hand, the **AdvERSEM**-trained GPT4o-mini demonstrates pronounced improvements in detecting these adversarial changes, par-

---

[3] 11 transformations were applied on all the 20 examples, in which 2 transformations, viz., *Polarity Negation Removal* and *Polarity Negation Addition* were applicable on 17 and 3 examples respectively, as only 3 out of 20 claims possessed a negation in their raw form.

Transformation	**Claim (original → transformed)**
Specificity: Topic Addition	In 2010, the population of Europe was larger than 61 → more than 61 million, according to demographic trends.
Specificity: Source Addition	The Woman in Black was abandoned by Hammer Film Productions in the 2010s in favor of working on Freddie vs. Jason , according to industry insiders.
Specificity: Subevent Addition	"Honeymoon" is the second major-label record by Elizabeth Woolridge Grant, and it was released in 2015.
ARG0-of	There exists a Korean band called Scandal is a band from Korea.
Hedging	People don't doubt that "Excuse My French" is the debut album of Karim Kharbouch(French Montana).
Focus Shift	There is not a natural element that goes by the name of Moscovium does not exist naturally.
Scalar Adverb Negation	Eurotas is definitely not a minor river of Laconia.
Polarity Neg Addition	The lead engineer of the iAPX 432 did not work at Intel for 20 more years → more than 20 years after its introduction.
Polarity Neg Removal	Dawood Ibrahim Kaskar was → is not from a place in Mira-Bhayandar, Thane district.
Scalar Negation	Exotic Birds hardly rejected to be → being an opening band for a band from Minneapolis.
Hedging	The general public doubts that Andrew Kevin Walker was born on August 14, 1864 and is a screenwriter.

Table 3: Transformation examples for each type on FEVER 2.0 claims – pieces of text removed from the original claim are shown in red while those added in the new claim are shown in green. The upper half shows Veracity Intact ones, while the bottom half shows Veracity Flipped ones.

	Manually Verified Evaluation Sets			Automatically Created Evaluation Sets					
	AdvERSEM-Test (T)		**FEVER 2.0 (R)**	**AVERITEC (R)**	**AVERITEC (T)**		**FEVER 2.0 (T)**		
Model	Macro	Micro	R	R	Macro	Micro	Macro	Micro	
Llama-3.1-8B	**.792**	**.766**	.652	.680	**.588**	.653	.498	.504	
gpt-4o-mini	.763	.762	.821	.786	.569	.642	.517	.550	
gpt4o	.754	.751	**.833**	**.880**	.585	**.675**	**.556**	**.579**	
gemini-2.0-flash	.761	.755	.731	.788	.562	.613	.515	.532	
gpt-4o-mini (Regular Trained)	.784	.793	.885	-	-	-	.569	.601	
gpt-4o-mini (**AdvERSEM** Trained)	**.850**	**.864**	**.897**	-	-	-	**.585**	**.625**	
llama-3.1-8B (Regular Trained)	.745	.749	**.805**	**.817**	.599	.652	.516	.526	
llama-3.1-8B (**AdvERSEM** Trained)	**.771**	**.779**	**.805**	.800	**.631**	**.676**	**.518**	**.532**	

Table 4: Performance on FEVER 2.0, AVERITEC, and **AdvERSEM-Test**. **R**=raw/original set, **T=AdvERSEM** transformed sets. Best scores across models are highlighted in bold. Top set of rows represent zero-shot variants.

Approach \ Label Change	Label Intact						Label Flipped					Overall	
	Topic	***Source***	***Subevent***	***ARG0-of***	***Hedging***	***Focus Shift***	***Scalar Adverb Neg***	***Polarity Neg Add***	***Polarity Neg Rem***	***Specific Scalar Negation***	***Hedging***	***Macro Average***	***Micro Average***
#Examples	20	20	20	20	20	20	20	17	3	20	20	200	200
llama-3.1-8B	.850	.737	.850	.750	.800	.789	.526	.588	.333	.500	.450	.652	.680
gpt-4o-mini	.900	.800	.900	.850	.900	.750	.684	.647	.667	.700	.600	.763	.773
gpt-4o	.900	.800	.900	.950	.900	.900	.526	.647	.667	.600	.500	.754	.763
gemini-2.0-flash	.850	.800	.850	.800	.850	.800	.667	.882	.667	.600	.600	.761	.767
llama-3.1-8B (Regular Trained)	.750	.700	.800	**.900**	**.800**	.650	.737	**.941**	.667	**.650**	.600	.745	.749
llama-3.1-8B (**AdvERSEM** Trained)	**.800**	**.750**	**.850**	.850	.750	**.850**	**.789**	.824	.667	.600	**.750**	**.771**	**.779**
gpt-4o-mini (Regular Trained)	**.950**	.850	.900	**.950**	**.900**	.750	.684	.824	.667	.550	.600	.784	.793
gpt-4o-mini (**AdvERSEM** Trained)	**.950**	**.900**	**.950**	.850	**.900**	**.800**	**.789**	**.941**	.667	**.800**	**.800**	**.850**	**.864**

Table 5: Fine-grained performance on **AdvERSEM-Test** demonstrating increased robustness.

ticularly in categories like *scalar adverb negation* (+10.5%), *polarity negation addition* (+11.7%), and *scalar negation* (+25%). Similar enhancements are observed with llama-3.1-8B, reinforcing that training with semantically structured adversarial examples notably boosts the robustness of models across various semantic alterations. These results indicate that structured AMR-based adversarial training generalizes well across different linguistic adversarial manipulations as well as significantly mitigates vulnerabilities to challenging alterations commonly employed in deceptive claims.

5.3 Analysis on FEVER and AVERITEC

We now look at the detailed performances over FEVER 2.0 and AVERITEC (shown in Appendices Table 10 and Table 11). We observe that the average robustness for all the zero-shot models is low. While utilizing FEVER 2.0 training data improves performance on the raw test set, it only gives a slight boost in robustness. When **AdvERSEM**-based examples are used for training, the robustness improves further, maintaining accuracy.

Across both FEVER 2.0 and AVERITEC, we observe a substantial drop in accuracy for all models on adversarially manipulated claims, particularly those that challenge deeper aspects of semantics (e.g., focus shift, structural reversal, or hedging). For example, **zero-shot GPT4o and gemini-2.0-flash experience significant drops in performance on manipulations like scalar adverb negation and polarity reversals**.

Zero-shot models, despite strong results on standard test sets, saw significant performance degradation on certain adversarial manipulations, especially those involving negations – e.g., GPT-4o's accuracy drops to 0.286 on scalar adverb negations.

On AVERITEC, which features noisy, real-world web claims, adversarially trained models retain accuracy even in the presence of more diverse and noisy evidence, showing the practical benefit of these techniques beyond synthetic testbed.

Proportion of Training Examples Used	Approach	Accuracy
0	gpt4o	.833
	gemini-2.0-flash	.731
	gpt-4o-mini	.821
1/3	gpt-4o-mini (FEVER 2 trained)	.800
	gpt-4o-mini (AdvERSEM Trained)	**.829**
1/2	gpt-4o-mini (FEVER 2 trained)	.859
	gpt-4o-mini (AdvERSEM Trained)	**.883**
1	gpt4o-mini (FEVER 2 trained)	.897
	gpt4o-mini (**AdvERSEM** Trained)	.863
	gpt4o-mini (AdvERSEM Trained EP)	**.897**

Table 6: Accuracy of different models trained on smaller subsets of FEVER 2.0 and evaluated on FEVER 2.0 Test split. Note that AMR-generated adversarial examples can improve raw performance with lesser number of annotated examples. (EP = Error Proportions)

5.4 Out-of-Domain Robustness

We also evaluate the out-of-domain robustness of these models by training on FEVER 2.0 and its **AdvERSEM** manipulations, then testing on the AVERITEC dataset.

Our results in Appendix Table 12 show that while zero-shot and regular finetuned models (such as GPT4o-mini and its FEVER 2-trained variant) maintain competitive accuracy on label-intact AMR manipulations, their robustness substantially declines on label-flipped and sentence-based adversarial attacks—especially for semantically challenging manipulations such as scalar negation or polarity reversal. Notably, adversarially finetuned models (i.e., those further trained on AMR-generated adversarial data) consistently outperform their non-adversarially trained counterparts across both AMR-based and sentence-level manipulations. For example, GPT4o-mini with AMR-based adversarial training achieves an average robustness of 0.674, outperforming the FEVER 2-trained (0.621) and vanilla (0.611) baselines, improving AMR robustness from 0.652 to 0.712. These results highlight that adversarial AMR-based training boosts LLM resilience to a broad spectrum of semantic manipulations, as well as generalizes well in out-of-domain settings.

5.5 AdvERSEM Training Sample Efficiency

We further demonstrate the effectiveness of incorporating **AdvERSEM**-based adversarial training in low-resource settings. We adjusted the amount of FEVER 2.0 training data available to the model and controlled the portion used for generating AMR adversarial examples—ranging from zero-shot (0) to one-third (1/3), half (1/2), and the full (1) dataset—and assessed performance on the FEVER 2.0 test set. Consistently, models enhanced with AMR-generated adversarial examples matched or exceeded the performance of those trained exclusively on human-annotated claims across all settings. The results are described in detail in Table 6.

The benefit of our approach is especially pronounced when labeled data is scarce. For example, GPT4o-mini trained with AMR-generated adversarial examples consistently outperforms its zero-shot counterpart, without signs of overfitting. In contrast, training with human-labeled FEVER 2.0 data alone leads to a 3% drop in performance when only one-third of the data is used. Moreover, expanding the proportion of AMR-based training data to one-third and half substantially closes the performance gap with the full-data baseline (0.897) by 11% and 82%, respectively. These findings underscore the sample efficiency of AMR-based augmentation, which introduces diverse and informative adversarial variations that enhance generalization, even with limited annotations.

6 Conclusions and Future Work

We introduced **AdvERSEM**, a novel framework that leverages Abstract Meaning Representations (AMR) to systematically evaluate and improve the robustness of groundedness evaluators through structured adversarial data augmentation. We uncover several vulnerabilities by introducing semantically controlled, fine-grained manipulations of claims. These adversarial manipulations expose specific weaknesses in existing evaluators and provide interpretable and actionable feedback beyond single-dimensional test scores, highlighting the importance of robustness and interpretability in groundedness evaluation, and also serve as a caution on agent-style modular systems, which predominantly rely on LLMs in a zero-shot manner.

Moreover, training groundedness evaluators using our structured adversarial examples significantly enhances their ability to withstand complex semantic perturbations such as hedging, negation, and specificity adjustments, addressing key vulnerabilities identified in LLMs as well as human writings. By generating challenging yet realistic training examples, our approach effectively reduces reliance on expensive annotated data, thereby facilitating efficient and robust model development. The **AdvERSEM** framework, and associated evaluation set **AdvERSEM-Test** can be readily extended to various other NLP applications, for systematically testing and enhancing model reliability.

7 Limitations

While **AdvERSEM** provides a structured and interpretable framework for systematically assessing and improving the robustness of groundedness evaluators, there are several limitations worth noting.

Our adversarial examples, though systematically designed and verified, are generated through a large language model (GPT-4o), which itself might introduce unintended biases or noise (Mitchell et al., 2025). Although we employed entailment checks and manual validation steps to mitigate these risks, some residual inaccuracies could persist. And hence, to provide a reliable estimate of the behavior of these groundedness evaluators, we manually modified split on top of the same.

AdvERSEM encompasses AMR parsing and AMR verbalization for English text. While AMR parsing has been expanded to many languages (Soto Martinez et al., 2024; Kang et al., 2024), cross-lingual and multi-lingual parsing is still an active area of research (Mansouri, 2025), and our method would need to be evaluated for those languages separately.

Additionally, while our method significantly improves robustness against specific adversarial manipulations, it might not cover all possible adversarial strategies, particularly those exploiting multi-sentence coherence or higher-level rhetorical manipulations. Future work could expand the diversity of manipulations and further explore the integration of human-generated adversarial examples to address these gaps comprehensively.

8 Ethics Statement

Our work operates within the broader context of combating misinformation, as large language models (LLMs) can be exploited for malicious purposes. Therefore, developing accurate, reliable, and robust assessment methods is essential. Systematically enhancing claim verification and groundedness evaluation is crucial for countering increasingly sophisticated misinformation and disinformation tactics, especially as LLM-generated simulations become proliferate (Dhole, 2024, 2023).

However, while recent advancements in LLM-based evaluations have shown promise, our research highlights that these models remain brittle and susceptible to exploitation. Consequently, research like ours, which is situated within fact verification and related domains, must always be supported by rigorous manual evaluation, particularly to ensure robustness. **AdvERSEM-Test**, is specifically created with this consideration in mind.

We used GPT4o and Grammarly to help improve the grammar of the text and for creating LaTeX outlines for tables and images.

Acknowledgments

This work was partially supported by the Microsoft Accelerate Foundation Models Award. The authors also thank the anonymous reviewers for their insightful comments.

References

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

Aimée Alonso-Reina, Robiert Sepúlveda-Torres, Estela Saquete, and Manuel Palomar. 2019. Team gplsi. approach for automated fact checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 110–114.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and verification over unstructured and structured information (feverous) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Kaustubh Dhole. 2023. Large language models as sociotechnical systems. In *Proceedings of the Big Picture Workshop*, pages 66–79.

Kaustubh Dhole. 2024. Kaucus-knowledgeable user simulators for training large language models. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 53–65.

Kaustubh Dhole and Eugene Agichtein. 2024. Llm judges for retrieval augmented argumentation.

Kaustubh Dhole, Kai Shu, and Eugene Agichtein. 2025. Conqret: A new benchmark for fine-grained automatic evaluation of retrieval augmented computational argumentation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5687–5713.

Kaustubh D Dhole. 2025. To retrieve or not to retrieve? uncertainty detection for dynamic retrieval augmented generation. *arXiv preprint arXiv:2501.09292*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Ken Hyland. 1998. Hedging in scientific research articles.

Jeongwoo Kang, Maximin Coavoux, Cédric Lopez, and Didier Schwab. 2024. Should cross-lingual AMR parsing go meta? an empirical assessment of meta-learning and joint learning AMR parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 43–51, Miami, Florida, USA. Association for Computational Linguistics.

Dongryeol Lee, Yerin Hwang, Yongil Kim, Joonsuk Park, and Kyomin Jung. 2025. Are LLM-judges robust to expressions of uncertainty? investigating the effect of epistemic markers on LLM-based evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8962–8984, Albuquerque, New Mexico. Association for Computational Linguistics.

Behrooz Mansouri. 2025. Survey of abstract meaning representation: Then, now, future. *arXiv preprint arXiv:2505.03229*.

Margaret Mitchell, Giuseppe Attanasio, Ioana Baldini, Miruna Clinciu, Jordan Clive, Pieter Delobelle, Manan Dey, Sil Hamilton, Timm Dill, Jad Doughman, et al. 2025. Shades: Towards a multilingual assessment of stereotypes in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11995–12041.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Ananjan Nandi, Christopher D Manning, and Shikhar Murty. 2025. Sneaking syntax into transformer language models with tree regularization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8006–8024, Albuquerque, New Mexico. Association for Computational Linguistics.

Amie Paige, Adil Soubki, John Murzaku, Owen Rambow, and Susan E. Brennan. 2024. Training LLMs to recognize hedges in dialogues about roadrunner cartoons. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 204–215, Kyoto, Japan. Association for Computational Linguistics.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36:65128–65167.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

William Soto Martinez, Yannick Parmentier, and Claire Gardent. 2024. Generating from AMRs into high and low-resource languages using phylogenetic knowledge and hierarchical QLoRA training (HQL). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 70–81, Tokyo, Japan. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *science*, 359(6380):1146–1151.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. Scifact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Peter RR White. 2003. Beyond modality and hedging: A dialogic view of the language of intersubjective stance. *Text & Talk*, 23(2):259–284.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI ’23, New York, NY, USA. Association for Computing Machinery.

A AMR Manipulations

In Table 7, we show our extended set of AMR manipulations. This list shows the remaining manipulations that may not guarantee veracity. For instance, in "*Quantifier Alteration*", when we change the scope or amount expressed in the claim, the new claim may not always maintain (or always flip) the veracity.

B Robustness Analysis on the Dev Set

In Table 9, we present the robustness analysis on the development set of FEVER 2.0 using GPT4o-mini. The fall in accuracy is calculated as the relative decline in accuracy when evaluated with the transformed dev set as compared to the raw dev set. We then use the inverse of the fall of accuracy, which is used to dictate the proportion of adversarial examples that would be used in training.

```
You are an advanced semantic parser. Read a sentence and
produce only its AMR or Abstract Meaning Representation,
without explanations.
Below is an example from the AMR 1.2.6 Specification.

EXAMPLE:
Sentence: Patrick Makau finished the marathon in 2 hours.
AMR:
(f / finish-01
    :ARG0 (p / person :wiki "Patrick_Makau_Musyoki"
           :name (n / name :op1 "Patrick" :op2 "Makau"))
    :ARG1 (r / run-02
        :ARG0 p
        :ARG1 (m / marathon)
        :duration (s2 / sum-of
            :op1 (t2 / temporal-quantity
                :quant 2
                :unit (h / hour)))))
...
...
Now convert the input sentence to AMR. Return only the AMR.
Sentence: {text}
AMR:
```

Figure 4: Text-to-AMR prompt example used for parsing original claims.

Figure 7 shows the robustness trends with and without adversarial examples.

C Sample Efficiency: Training with limited data

The Figure 7 illustrates the sample efficiency of our approach, highlighting its robustness across varying numbers of original examples.

D Prompt Templates

We provide the prompt templates used in each step of our framework. Figure 4, 5, and 6 correspond to the key steps: AMR parsing, AMR manipulation, and generated example validation, respectively.

```
The following is an AMR or Abstract Meaning Representation
of a claim {claim} which is {label} by evidence:

Please perform the following action to create a new AMR
whose verbalisation would be different:

The action should make the new AMR represent a new natural
looking, grammatical, and sensible claim, and also
different from the
original claim. If there is no previous AMR,
then only return
'NO CHANGE'.

Return the new AMR.
Previous AMR: {AMR}
Label: {The new Label}
Description: {Description of Manipulation}
Action: {Action Name of the Manipulation}

New AMR:
```

Figure 5: Prompt template for AMR manipulation based on graph operations and label changes.

```
"The following two statements are intended to
not contradict each other. "
"Check if they are logically consistent and
do not contradict each other.\n\n"
"Original statement: {}
\n\nNew statement: {}\n\n"
"Respond with 'True' if they do not contradict each other, "
"or 'False' if they do contradict.
Do not respond with anything else."

----------------------------------------------------------------

"The following two statements are expected to reflect
contradictory positions. "
"Check if they indeed contradict each other.\n\n"
"Original statement: {}
\n\nRevised statement: {}\n\n"
"Respond with 'True' if they contradict each other, "
"or 'False' if they do not contradict. Do not respond with
anything else."
```

Figure 6: Entailment and Contradiction

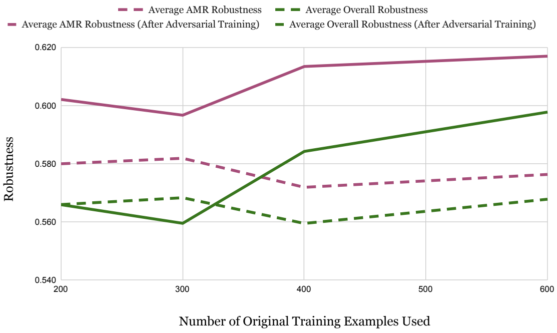


Figure 7: Robustness Trends of With and Without Including Adversarial Examples. At each point, the adversarial examples are generated for the same number of original examples (x-axis)

Manipulation	Description
Modality Shifting	Weaken or strengthen the speaker's commitment to the claim. (e.g., Replace "must" with "might".)
Quantifier Alteration	Change the scope or amount expressed in the claim. (e.g., Alter "100" to "10".)
Presupposition Removal	Delete or contradict a background assumption (e.g., Remove an assumed ongoing subevent.)
Implicature Disruption	Break implied event ordering or assumptions of contrast. (e.g., Replace causal relation with flat conjunction "and")
Rhetorical Question Framing	Transform into a rhetorical question that implies doubt. (e.g., Change to "Did it really..")
Parataxis vs Hypotaxis	Change logical or causal relationships to a flat structure, altering implications. (e.g., Replace "because" with "and")
Figurative Interpretation	Replace with metaphorical or sarcasm (e.g., Replace "bulldoze" with "dominate" to imply sarcasm.)

Table 7: Adversarial AMR manipulations. These manipulations do not guarantee the veracity of the transformed claim.

	Dataset	**AMR Transformations (AdvERsem-FEVER) (Automatic)**				**AMR Transformations (AdvERsem-AVERITEC) (Automatic)**				**AdvERSem-Test (Manual)**
	Split	**Train**		**Dev**		**Train**		**Dev**		**Test**
		TA (%)	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**
Label Intact	Specificity: Topic Addition	99.84	68.90	100	79.75	100.00	80.96	100.00	84.40	100
	Specificity: Source Addition	100.00	71.61	100	81.01	99.85	78.15	100.00	81.65	100
	Specificity: Subevent Addition	99.84	64.75	100	69.62	99.62	77.62	100.00	72.48	100
	Structure Reversal	100.00	67.46	98.73	70.89	99.85	71.47	100.00	72.48	100
	Hedging	99.84	75.92	100	77.22	100.68	77.62	99.08	76.15	100
	Focus Shift	99.84	67.94	100	69.62	100.08	74.81	100.00	71.56	100
	Scalar Adverb Negation	99.84	25.20	100	36.71	100.23	33.54	101.83	25.69	100
Label Flipped	Polarity (Negative Addition)	68.42	7.18	70	7.59	85.43	6.68	63.30	2.75	85
	Polarity (Negative Removal)	30.30	3.35	28	3.80	12.75	2.73	14.68	1.83	15
	Adding Other Scalar Negations	99.52	37.16	100	39.24	100.08	27.62	101.83	33.94	100
	Hedging	100.00	63.48	100	67.09	100.76	62.82	100.92	58.72	100

Table 8: Transformation Applicability of each transformation without (TA) and with verifier (TAv).

Manipulation Type	**Veracity Change**	**Manipulation Category**	**Fall in Accuracy**
	Raw		- (.810)
AMR Based	**Veracity Intact**	Specificity: Topic Addition	7.91 (.746)
		Specificity: Source Addition	20.92 (.641)
		Specificity: Subevent Addition	7.98 (.745)
		Structure Reversal	11.83 (.714)
		Hedging	12.99 (.705)
		Focus Shift	5.74 (.764)
	Veracity Flipped	Scalar Adverb Negation	31.90 (.552)
		Polarity Negation (Addition)	17.71 (.667)
		Polarity Negation (Remove)	58.85 (.333)
		Scalar Negation	24.34 (.613)
		Hedging	27.80 (.585)
Sentence Based	**Veracity Intact**	Entity Attribute Substitution	25.06 (.607)
		Temporal Numerical Substitution	29.46 (.571)
	Veracity Flipped	Entity Attribute Substitution	31.72 (.553)
		Temporal Numerical Substitution	64.73 (.286)

Table 9: "Fall in Accuracy" indicates the difference from 0.810, with the actual accuracy shown in parentheses. *Fewer examples were available for these negation-based transformations.

```
Classify whether the given evidence SUPPORTS or REFUTES
the given claim.

Evidence: {wiki_text}

Claim: {claim}

Answer (SUPPORTS or REFUTES):
```

Figure 8: Fact Verification instruction used to evaluate the veracity of the claims

		AMR Based Transformations												Sentence Transformations					
Approach \ Label Change		Label Intact						Label Flipped						Label Intact		Label Flipped			
	FEVER	***Topic***	***Source***	***Subevent***	***ARG0-of***	***Hedging***	***Focus Shift***	***Scalar Adverb Neg***	***Polarity Neg Add***	***Polarity Neg Rem***	***Specific Scalar Negation***	***Hedging***	***AMR Average***	***Entity Attribute***	***Temp-Numer***	***Entity Attribute***	***Temp-Numer***	***Sentence Average***	***Overall Average***
#Examples	78	60	59	57	52	62	59	21	21	50	31	49	-	54	58	44	42	-	-
gpt4o	.833	.717	.661	.702	.692	.613	.831	.286	**.500**	.000	.484	.510	.545	.630	**.500**	.500	**.714**	**.586**	.556
gemini-2.0-flash	.731	.672	.586	.625	.635	.610	.746	.429	.333	.000	.516	**.633**	.526	.612	.300	.452	.571	.485	.515
gpt-4o-mini	.821	.717	.678	.719	.731	.629	.831	.429	.000	.000	**.581**	.449	.524	.556	.300	.568	.571	.499	.517
llama-3.1-8B (FEVER 2 Trained)	**.702**	.674	**.609**	**.725**	**.634**	.568	.667	.500	.200	.000	.381	**.643**	.509	.486	.250	**.588**	.667	.498	.506
llama-3.1-8B (**AdvERSEM** Trained)	.698	**.702**	.587	.718	**.634**	**.581**	**.689**	**.643**	**.200**	.000	**.429**	**.677**	**.533**	.486	.250	.500	.667	.476	**.518**
gpt-4o-mini (FEVER 2 trained)	.885	.800	.695	.719	.846	.710	**.881**	.381	.333	.000	.484	.490	.576	.660	.400	.545	.571	.544	.568
gpt-4o-mini (**AdvERSEM** Trained)	**.897**	**.817**	**.746**	**.842**	**.827**	**.726**	.847	**.429**	.167	.000	**.548**	**.551**	**.591**	**.796**	**.500**	.409	.571	**.569**	**.585**

Table 10: Accuracy of different models on transformed adversarial test splits of FEVER 2.0 (Thorne et al., 2018a). Best scores across models are highlighted in bold.

		AMR Based Transformations												Sentence Transformations					
Approach \ Label Change		Label Intact						Label Flipped						Label Intact		Label Flipped			
	AVERITEC	***Topic***	***Source***	***Subevent***	***ARG0-of***	***Hedging***	***Focus Shift***	***Scalar Adverb Neg***	***Polarity Neg Add***	***Polarity Neg Rem***	***Specific Scalar Negation***	***Hedging***	***Average AMR***	***Entity Attribute***	***Temp-Numer***	***Entity Attribute***	***Temp-Numer***	***Sentence Average***	***Overall Average***
#Examples	120	91	89	83	68	89	76	24	33	81	36	63	-	90	70	47	56	-	-
llama-3.1-8B	.766	.765	.813	.773	.789	.684	.738	.250	.375	.667	.226	**.327**	.583	.859	.793	.386	.375	.603	**.588**
gpt-4o-mini	.786	.807	.818	.775	.773	.816	.716	**.333**	.000	.667	**.457**	.290	**.587**	.841	.733	.311	.200	.521	.569
gpt4o	**.880**	**.839**	**.864**	**.863**	**.785**	**.826**	**.811**	.043	.125	.667	.235	.290	.577	**.852**	**.833**	**.444**	.300	**.608**	.585
gemini-2.0-flash	.788	.761	.741	.800	.781	.753	.736	.235	**.500**	.333	.278	.279	.563	.764	.655	.438	**.375**	.558	.562
llama-3.1-8B (Averitec Trained)	**.817**	**.769**	.775	.771	**.721**	.719	.724	.458	.250	.667	.361	.238	.587	.767	**.839**	.319	**.300**	.556	.579
llama-3.1-8B (**AdvERSEM** Trained)	.800	.747	**.787**	**.819**	.676	**.798**	**.776**	**.625**	**.375**	.667	**.639**	**.683**	**.690**	**.789**	.806	**.447**	.200	**.561**	**.656**

Table 11: Accuracy of different models on transformed adversarial test splits of AVERITEC dataset (Schlichtkrull et al., 2023). Best scores across models are highlighted in bold.

	AMR Based Transformations											Sentence Transformations					
Approach \ Label Change	Label Intact						Label Flipped					Label Intact		Label Flipped			
	Topic	***Source***	***Subevent***	***ARG0-of***	***Hedging***	***Focus Shift***	***Scalar Adverb Neg***	***Polarity Neg***	***Specific Scalar Negation***	***Hedging***	***AMR Robustness***	***Entity Attribute***	***Temp-Numer***	***Entity Attribute***	***Temp-Numer***	***Sentence Robustness***	***Average Accuracy***
gpt-4o-mini	.814	.823	.761	.752	.798	**.770**	.407	.468	.386	.319	.630	**.836**	.734	.316	.356	.561	.611
gpt-4o-mini (FEVER 2 trained)	**.864**	**.886**	**.850**	**.818**	**.816**	.743	.292	.375	.486	.387	.652	.795	**.800**	**.356**	.200	.538	.621
gpt-4o-mini (AdvERSEM Trained)	.818	.830	.813	.788	.724	.757	**.583**	**.500**	**.629**	**.677**	**.712**	.818	.767	.289	**.400**	**.568**	**.674**

Table 12: Accuracy of **AdvERSEM** trained models on Out-of-domain data, with FEVER 2.0 adversarial examples and evaluated on AVERITEC test split (Schlichtkrull et al., 2023), broken down by AMR manipulation type, highlighting robustness to out-of-domain adversarial variations.

Connecting Concept Layers and Rationales to Enhance Language Model Interpretability

Thomas Bailleux,[1], Tanmoy Mukherjee[1], Pierre Marquis[1], Zied Bouraoui[1]
[1] CRIL, Univ. Artois & CNRS, France
`{bailleux,mukherjee,marquis,bouraoui}@cril.fr`

Abstract

With the introduction of large language models, NLP has undergone a paradigm shift where these models now serve as the backbone of most developed systems. However, while highly effective, they remain opaque and difficult to interpret, which limits their adoption in critical applications that require transparency and trust. Two major approaches aim to address this: rationale extraction, which highlights input spans that justify predictions, and concept bottleneck models, which make decisions through human-interpretable concepts. Yet each has limitations—rationales lack semantic abstraction while concepts miss fine-grained linguistic grounding. Crucially, current models lack a unified framework that connects where a model looks (rationales) with why it makes a decision (concepts). We introduce CLARITY, a model that first selects key input spans, maps them to interpretable concepts grounded in linguistic semantics, and then predicts using only those concepts. This design reveals how surface-level linguistic patterns map to abstract semantic representations, supporting faithful, multi-level explanations and allowing users to intervene at both the rationale and concept levels. CLARITY achieves competitive accuracy while offering improved transparency and semantic interpretability. The source code can be accessed at this link: CLARITY.

1 Introduction

Language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have transformed NLP, forming the basis of many systems and excelling in various tasks such as sentiment analysis and document classification. Despite their impact, these models are black boxes, with complex, opaque outputs, posing challenges in sectors where transparency is crucial, like healthcare and law. To address interpretability, NLP has focused on two main approaches: *rationale extraction* and *concept bottleneck models*. Rationale extraction identifies input text segments that justify model predictions Lei et al. (2016), promoting interpretability by highlighting essential evidence while using sparsity and regularization for accuracy Paranjape et al. (2020). These explanations, however, are often limited to token-level insights and lack broader semantic context. In contrast, concept bottleneck models (CBMs) encourage the model to make predictions via interpretable intermediate representations, often aligned with human-defined concepts (Koh et al., 2020). CBMs offer several advantages, including the ability to intervene on model reasoning and support post-hoc debugging. However, these models typically assume that the relevant concepts are already known or provided, and they operate on whole examples, without leveraging fine-grained input regions that support those concepts. As a result, they often lack fine-grained textual grounding, making it unclear where in the input a concept arises. Despite their complementary strengths, rationale extraction and CBMs have largely evolved in isolation. Rationale-based approaches offer textual grounding but lack semantic abstraction, while CBMs provide interpretable reasoning structures without linking them to specific input regions. This separation is particularly problematic for semantic understanding, where surface forms and deep meanings are intrinsically connected. Understanding *how* linguistic expressions contribute to *what* semantic concepts is crucial for advancing interpretable semantic processing. Crucially, current models lack a unified framework that connects where a model looks (rationales) with why it makes a decision (concepts).

To bridge this gap, we propose CLARITY, a unified framework that tightly integrates fine-grained rationale extraction with concept-based reasoning. By explicitly mapping selected input spans to intermediate concepts before classification, CLARITY produces multi-level explanations that are both

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 409–429
November 8-9, 2025 ©2025 Association for Computational Linguistics

textually grounded and semantically meaningful. This approach directly addresses a key challenge in computational semantics: understanding how surface linguistic patterns realize abstract semantic categories. Specifically, CLARITY decomposes prediction into a three-stage process: (1) it identifies sparse, contiguous rationales from the input that correspond to semantically coherent units; (2) it maps these rationales to a low-dimensional vector of interpretable concepts that capture semantic dimensions; and (3) it predicts the final output label using only these activated concepts. The contributions we have made can be described as follows:

- An architecture that unifies rationale extraction and CBMs for multi-level interpretability.
- A sparse-attention-based rationale extractor that selects concise, coherent evidence spans.
- A concept bottleneck layer that supports concept interventions and semantic abstraction.
- Extensive empirical validation across five datasets, demonstrating competitive performance and faithful, structured explanations.

2 Related Works

Interpretability in NLP has advanced rapidly in recent years, moving from simple feature attributions to structured, multi-level interpretability frameworks. We review major work on rationale extraction, CBMs, and hierarchical interpretability.

Rationale Extraction Rationale-based methods aim to identify input spans that are sufficient to justify predictions. It has advanced significantly since (Lei et al., 2016) and the information bottleneck from (Paranjape et al., 2020). UNIREX (Chan et al., 2022) offers a unified learning framework balancing faithfulness, plausibility, and performance, with a 32.9% improvement in Normalized Relative Gain across five datasets. FiD-Ex (Lakhotia et al., 2021) addresses sequence-to-sequence model issues by using sentence markers to encourage extractive explanations. REFER (Ghasemi Madani and Minervini, 2023) created a rationale extraction framework with a differentiable extractor to enhance task and explanation fidelity through concurrent training. Recent work increasingly blends causal reasoning with rationale extraction, addressing confounding factors in rationale models (Ghoshal et al., 2022). However, these models operate purely at the token level, lacking abstraction or semantic generalization.

Concept Bottleneck Models CBMs guide predictions through a bottleneck of human-interpretable concepts, first proposed by Koh et al. (2020) for image classification and later adapted to NLP. Text Bottleneck Models (Ludan et al., 2023) use CBMs for text classification, providing global and local explanations via LLMs discovering concepts without human input. CB-LLMs (Sun et al., 2025) introduced inherently interpretable neurons in LLMs for text tasks, aligning neuron activations with concept scores for classification and combining interpretable and unsupervised neurons for generation. CT-LLMs (Bhan et al., 2025) resolved concept completeness and classification leakage by generating concept labels unsupervised with small language models, removing the need for predefined concepts. While CBMs provide global interpretability and allow concept-level interventions, they typically assume concept supervision and operate on entire inputs, making them less suitable for tasks requiring localized justifications.

Hierarchical Interpretability Methods Several methods attempt to bridge the gap between local explanation mechanisms (rationales) and overarching global explanations (concepts). HEDGE (Chen et al., 2020) was the first to introduce hierarchical explanations for text classification by detecting feature interactions. Instead of solely highlighting key tokens, HEDGE illustrates how words and phrases combine across different hierarchical levels, effectively connecting token-level details to broader conceptual insights. T-EBAnO (Ventura et al., 2021) provides explanations specific to predictions by identifying impactful text regions and offering model-wide explanations through the aggregation and examination of these local insights across various inputs. This framework links specific parts of the input to overarching patterns seen throughout the dataset. HINT (Yan et al., 2022) shifted the focus of model interpretation from individual words to topics as core semantic components, constructing a hierarchical topic structure for explaining decisions across different abstraction levels. It has shown competitive performance with leading text classifiers while offering more easily comprehensible explanations. The intersection of interpretability and semantics has received limited attention despite its importance. Prior work on semantic role labeling (Palmer et al., 2005) and frame semantics (Fillmore and Baker, 2001) provides theoretical grounding for connecting surface forms to semantic concepts, but these approaches typically operate independently of neural interpretability methods. Our work bridges this gap by operationalizing semantic interpretability within neural architectures, enabling empirical investigation of how models learn linguistic-semantic mappings. InterroLang (Feldhus et al., 2023) enables users to engage in interactive dialogue to explore various explanation levels through natural language. This methodology combines feature attribution with conceptual explanations, allowing for flexible exploration across several interpretability tiers.

Positioning. Our framework, CLARITY, unifies rationale extraction and CBMs in a single architecture, enabling a more integrated and controllable form of interpretability. While prior rationale extraction methods such as UNIREX, FiD-Ex, and REFER focus on improving plausibility, fabrication avoidance, or end-to-end differentiability, they operate solely at the token level. In contrast, CLARITY introduces a rationale-guided concept mapping mechanism, where selected spans directly influence the activation of interpretable concepts. This ensures that concept representations are grounded in meaningful evidence, aligning semantic reasoning with input-level justifications. Conversely, existing CBMs such as Text Bottleneck Models and CT-CBMs often discover or annotate concepts independently of specific inputs, relying on LMs or latent clustering. CLARITY addresses this by incorporating concept-constrained rationale extraction, where activated concepts inform and refine the selection of rationales. This bidirectional interaction creates more coherent and semantically enriched explanations than approaches that treat concept prediction and span selection as separate tasks. Furthermore, frameworks for multi-level interpretability like HEDGE, T-EBAnO, and HINT connect local and global signals via feature interactions or topic aggregation. CLARITY goes further by learning explicit hierarchical attention between token-level rationales and high-level concepts, producing structured, end-to-end explanations across abstraction levels.

Positioning Against LLM-based Explanation Methods Recent work has explored using large language models for generating post-hoc explanations through prompting (Wiegreffe et al., 2021; Lampinen, 2022). While valuable, these approaches serve a fundamentally different purpose than our inherently interpretable architecture:

- **Inherent vs. Post-hoc Interpretability:** CLARITY builds interpretability into the model architecture, ensuring explanations directly reflect the decision process. LLM-based explainers generate separate explanations that may not accurately represent the original model's reasoning.

- **Computational Efficiency:** Our framework provides explanations without additional LLM calls, making it suitable for real-time applications. Prompting-based methods require expensive LLM inference for each explanation.

- **Controllability:** CLARITY enables causal interventions at both rationale and concept levels. Post-hoc explanations typically don't support direct model manipulation.

These approaches are complementary rather than competing solutions, addressing different interpretability requirements across the ML deployment pipeline.

3 Methodology

We introduce CLARITY, a framework that combines rationale extraction with concept bottleneck mechanisms. This section formalizes the task and presents the model architecture, interpretability constraints, training objective and intervention procedure.

3.1 Problem Formulation

Given a tokenized input sequence $X = (x_1, \ldots, x_n) \in \mathcal{V}^n$ from vocabulary $\mathcal{V}$, the task is to predict a class label $y \in \mathcal{Y}$. CLARITY computes three intermediate representations

- *Token Embeddings:* $H = (h_1, \ldots, h_n) \in \mathbb{R}^{n \times d}$, produced by a frozen or finetuned encoder (e.g., BERT (Devlin et al., 2019)).

- *Rationale Mask:* $R \in \{0, 1\}^n$, where $R_i = 1$ indicates that token x_i is part of the extracted rationale (Lei et al., 2016).

- *Concept Vector:* $C \in [0, 1]^m$, where c_j denotes the activation strength of concept j (Koh et al., 2020).

CLARITY decomposes prediction into interpretable intermediate steps. Let $H \in \mathbb{R}^{n \times d}$ denote token embeddings for an input sequence $X \in \mathcal{V}^n$. First, the *rationale selector* $g_\eta : \mathcal{V}^n \to \{0,1\}^n$ produces a binary mask R where $R_i = 1$ indicates token x_i is selected. The *concept mapper* $h_\phi : \{0,1\}^n \times \mathbb{R}^{n \times d} \to [0,1]^m$ then maps selected tokens to a concept vector $C \in [0,1]^m$, with each dimension c_j representing a human-interpretable feature. Finally, the *classifier* $k_\psi : [0,1]^m \to \mathcal{Y}$ predicts the label $\hat{y}$ from C.

3.2 Model Architecture

To produce interpretable predictions, CLARITY follows a modular design that decomposes decision-making into three distinct stages: rationale selection, concept abstraction, and label prediction. Figure 1 illustrates the process. Notice that each stage is implemented as a dedicated component, enabling explicit control and transparency at multiple levels of the model's reasoning process. Formally, CLARITY is structured as a three-stage pipeline where f_θ can be instantiated as either a language model (LM) or a simpler classifier, depending on the task:

$$f_\theta(X) = k_\psi\left(h_\phi\left(g_\eta(X), H\right)\right) \tag{1}$$

Rationale selector. It predicts a binary mask $R \in \{0,1\}^n$ over input tokens, identifying the subset deemed relevant for the final prediction. This component highlights specific spans of text that drive the model's decision, making the process more transparent.

Concept mapper. It transforms the selected rationale (in combination with token embeddings H) to a low-dimensional, interpretable concept vector $C \in [0,1]^m$. This crucial interpretability layer bridges the gap between low-level text features and high-level decisions by: (i) Encoding the pooled rationale representation through a lightweight neural network; (ii) modeling explicit concept interactions through a learnable symmetric matrix; (iii) Enforcing sparsity to ensure only relevant concepts activate for each input; (iv) Encouraging diversity to prevent redundancy between learned concepts; and (v) Enabling concept interventions for causal analysis of model behavior.

Classifier. It makes a prediction in the label space $\mathcal{Y}$ using only the concept vector (or optionally combining it with raw encoder representations through a skip connection). This final stage creates a direct link between human-interpretable concepts and model decisions.

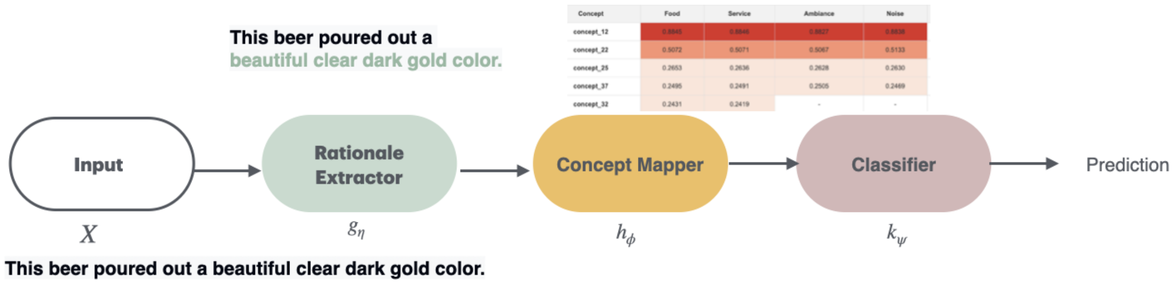
This modular structure enables CLARITY to generate transparent and controllable predictions by separating information selection (through rationales), semantic abstraction (through concepts), and decision-making (through classification). The concept mapper h_ϕ in particular serves as the critical "bottleneck" in this architecture, ensuring that predictions pass through a human-interpretable semantic space before reaching the final output.

3.3 Interpretability Constraints

To guide the model toward producing faithful and human-aligned explanations, we introduce a set of structural constraints on both the rationale and concept representations.

Rationale Constraints. To guarantee that extracted rationales are both meaningful and succinct, we impose three constraints on the rationale mask $R \in \{0,1\}^n$. First, we enforce *contiguity*, where rationales must form continuous spans $R_i = R_k = 1$ and $i < j < k \Rightarrow R_j = 1$. This encourages the model to select coherent phrases rather than disjoint tokens. Second, we promote *sparsity* by constraining the number of selected tokens. Namely, only a small fraction of tokens is selected: $\|R\|_1 \leq \tau n$ where $\tau \in (0,1)$ is hyperparameter and n is the sequence length. This prevents the model from defaulting to copying the full input. Finally, we require *faithfulness*, meaning that predictions based on R should approximate predictions based on the full input: $P(Y|X,R) \approx P(Y|X)$ (DeYoung et al., 2020).

Concept Constraints. To maintain semantic clarity and interpretability in the concept layer, we introduce constraints on the concept vector $C \in [0,1]^m$ and the concept decoder. First, we encourage *non-redundancy* by promoting orthogonality among concept embeddings: $\max_{j \neq k} \langle w_j, w_k \rangle \leq \epsilon$. This encourages each concept to capture a distinct semantic dimension. We also apply a *sparsity* constraint on the concept vector itself, enforcing $\|C\|_0 \leq \kappa$ where κ controls the maximum number of concepts active per example. Finally, we promote *atomicity* by signifying each c_j corresponds to a human-interpretable semantic unit (Koh et al., 2020). Implicitly, atomicity is promoted through the integration of sparsity, orthogonality and the



This beer poured out a beautiful clear dark gold color.

This beer poured out a beautiful clear dark gold color.

Concept	Food	Service	Ambiance	Noise
concept_12	0.8845	0.8846	0.8827	0.8838
concept_22	0.5072	0.5071	0.5067	0.5133
concept_25	0.2653	0.2636	0.2628	0.2630
concept_37	0.2495	0.2491	0.2505	0.2469
concept_32	0.2431	0.2419	-	-

Figure 1: Overview of the CLARITY architecture that operates in three interpretable stages: (1) rationale extraction selects a sparse subset of input tokens relevant for the prediction; (2) the concept mapper projects these into a low-dimensional, interpretable concept space; and (3) the classifier predicts the output label using only the activated concepts.

alignment of concept activation with specific input regions using the rationale extractor.

3.4 Training Objective

We use a composite objective that balances task performance with interpretability. The total loss is a weighted sum of four components:

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{cont}}\mathcal{L}_{\text{contiguity}} + \lambda_{\text{div}}\mathcal{L}_{\text{diversity}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparsity}} \quad (2)$$

Where each component in the loss fulfills a goal related to either interpretability or performance. Specifically,

Classification loss $\mathcal{L}_{\text{cls}}$: A standard cross-entropy loss that measures how accurately the model predicts the target label y given the final concept representation C. This term ensures task performance is preserved.

Contiguity loss $\mathcal{L}_{\text{contiguity}}$: Encourages the rationale mask R to consist of smooth, contiguous spans rather than scattered tokens. It is computed as the sum of absolute differences between adjacent binary rationale values; $\sum_{i=1}^{n-1} |R_{i+1} - R_i|$.

Diversity loss $\mathcal{L}_{\text{diversity}}$: Promotes orthogonality between concept embeddings by minimizing the deviation of $WW^\top$ from the identity matrix, where W is the concept decoder's weight matrix. This reduces redundancy between learned concepts.

Sparsity loss $\mathcal{L}_{\text{sparsity}}$: Penalizes overly dense rationale masks and concept activations. The first term enforces that the rationale covers approximately a target fraction τ of the input sequence. The second encourages the concept vector C to be sparse (i.e., few concepts should be active).

Algorithm 1 in the Appendix A summarizes the training procedure.

3.5 Concept Intervention Procedure

To evaluate the causal role of learned concepts in model predictions, we conduct targeted *concept interventions*. This technique modifies the activation of specific intermediate concepts to observe the effect on downstream predictions, offering insight into model behavior. Given a trained model and input example x, we first extract the concept vector $\mathbf{c} = \texttt{ConceptMapper}(x)$ and the original prediction $\hat{y} = \arg\max f(\mathbf{c})$. For a target concept index i, we replace c_i with a new value $c'_i \in [0, 1]$, producing an intervened vector $\mathbf{c}'$ where:

$$c'_j = \begin{cases} c'_i, & \text{if } j = i \\ c_j, & \text{otherwise} \end{cases}$$

The updated prediction $\hat{y}' = \arg\max f(\mathbf{c}')$ reflects the impact of this intervention. We perform both *zeroing* ($c'_i = 0$) and *maximization* ($c'_i = 1$) interventions to assess each concept's necessity and sufficiency, respectively. The difference in output probabilities $\Delta p = f(\mathbf{c}') - f(\mathbf{c})$ quantifies the influence of the concept on the prediction. To control for interactions, we optionally freeze the skip connection (if enabled) during this process, isolating the concept pathway. This analysis helps identify which concepts act as decision bottlenecks and which are spurious or redundant.

3.6 Explanation Pipeline

At inference time, CLARITY generates explanations by passing inputs through a three-stage pipeline. This process mirrors the model architecture and reflects the interpretability built into each component. First, the rationale extractor selects a sparse, contiguous subset of tokens from the input sequence $R_i > 0.5$. Next, the selected rationale is passed to the concept mapper, which transforms the span-specific embeddings into a compact, interpretable concept vector. Concepts with activation scores above a threshold $c_j > \alpha$ serve as a semantic abstraction of the input. Finally, the classifier predicts the output label based on activated concepts to compute the label: $\arg\max_{y \in \mathcal{Y}} k_\psi(C)_y$.

4 Experiments

To evaluate the effectiveness of CLARITY, we conduct experiments on five diverse text classification tasks. Our analysis focuses on both predictive performance and interpretability, examining how well the model maintains accuracy while generating faithful, semantically meaningful explanations. Section 5 provides detailed ablations and rationale quality analyses.

4.1 Experimental Setting

Datasets. We evaluate our model on selected datasets with varying characteristics to ensure the generalizability, including CEBaB (Abraham et al., 2022), SST-2 (Socher et al., 2013), AG News (Zhang et al., 2015), Yelp Polarity (Zhang et al., 2015), and DBpedia (Lehmann et al., 2015). See Appendix B for details.

Implementation. Our classification experiments utilize a pre-trained DistilBERT-base-uncased model (Sanh et al., 2019) as the encoder backbone, chosen for its balance of efficiency and performance. Other LMs are also used for comparison. We adopted a unified training framework across all datasets, with hyperparameters tuned for scalability and stability. To manage large datasets efficiently, we incorporated techniques such as mixed-precision training (FP16) and gradient accumulation. Appendix C details all training details.

Evaluation Metrics Initial performance is evaluated by utilizing accuracy for classification tasks. For interpretability, we also extract rationales and concept activations to inspect decision pathways.

Table 2 reports results on classification. Our model achieved competitive performance across all datasets on classification tasks. The model performed particularly well on the DBpedia dataset, suggesting that topic classification benefits more from the concept bottleneck approach than sentiment analysis tasks. This aligns with our hypothesis that well-defined topic categories map more cleanly to interpretable concepts. Further analysis is provided in Appendix D Table 4.

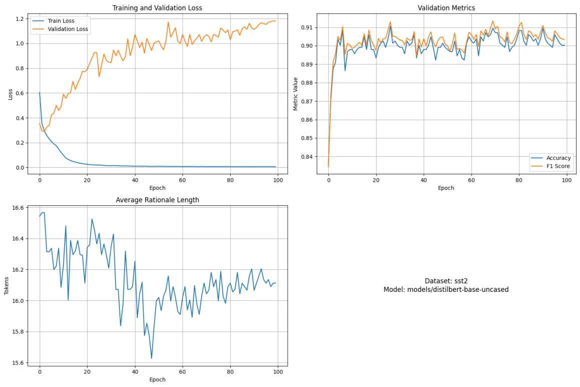


Figure 2: SST2 training dynamics

5 Ablation Analysis

To better understand the internal behavior of CLARITY and validate the design choices in its architecture, we conduct a series of ablation studies. We begin with an analysis of training dynamics and follow with targeted evaluations of rationale behavior, concept interventions, and architectural variants. Additional analyses are provided in the Appendix.

5.1 Training Dynamics

We analyze training dynamics using metrics loss, accuracy, F1 score, and rationale span characteristics. On SST-2, training loss decreased towards zero, while validation loss increased as shown in Figure 2, indicating mild overfitting without affecting stable and high validation accuracy. Accuracy and F1 score quickly improved to about 90% within 10 epochs, maintaining this balanced performance across classes. Rationale length initially varied between 15.4 and 16.5 tokens as the model explored strategies, then stabilized around 16 tokens after epoch 60, suggesting a reliable strategy for choosing informative segments. Additional analyses are provided in Appendix C.5.

5.2 Rationale Quality Analysis

We evaluated the quality of rationales extracted by different configurations of our CLARITY to understand how architectural choices and hyperparameters affect explanation quality. Our comprehensive evaluation methodology and detailed experimental setup are presented in Appendix E. Here, we summarize the key findings of our analysis, which focused on automated metrics including faithfulness (agreement between predictions using rationale-only vs. full text), contiguity (average

Table 1: Comprehensive Performance Comparison Across Model Architectures and Interpretability Methods

Method	Backbone	Params	Interpretable	AG News	DBpedia	CEBaB	Yelp	Avg
Black-box Baselines								
BERT-base	BERT-base	110M	✗	91.0	99.4	78.9	96.2	91.2
RoBERTa-large	RoBERTa-large	355M	✗	92.3	99.6	82.1	97.8	92.9
Interpretable Methods								
LIME	BERT-base	110M	✓	89.2	97.8	76.3	94.1	89.4
SHAP	BERT-base	110M	✓	89.8	98.1	77.1	94.8	90.0
C³M	BERT	110M	✓	91.5	99.5	79.3	95.8	91.3
CB-LLM	BERT	110M	✓	90.0	99.3	76.5	95.0	90.1
CLARITY (Multiple Backbones)								
CLARITY	DistilBERT	66M	✓	90.6	99.3	78.4	96.0	90.9
CLARITY	BERT-base	110M	✓	90.8	99.4	79.1	96.2	91.1
CLARITY	RoBERTa-base	125M	✓	91.1	99.5	79.8	96.5	91.7
CLARITY	BERT-large	340M	✓	91.9	99.6	80.5	97.1	92.3
CLARITY	RoBERTa-large	355M	✓	92.3	99.7	81.2	97.8	92.8

Table 2: Classification performance comparison across models and datasets (Accuracy in %)

Model	Interpretable	Backbone	AG News	DBpedia	CEBaB	Yelp Polarity	SST-2	Avg
BERT-base	✗	–	91.0	99.4	78.9	96.2	90.7	91.2
DeBERTa-large	✗	–	92.0	99.4	83.2	97.3	93.4	93.1
GPT-3.5 (fine-tuned)	✗	–	91.6	99.2	82.0	97.1	92.7	92.5
GPT-4 (10-shot)	✗	–	92.3	99.5	83.8	97.8	94.1	93.5
Naive Bayes	✓	–	84.0	96.5	71.2	91.4	81.5	84.9
C³M	✓	BERT	91.5	99.5	79.3	95.8	90.2	91.3
CB-LLM	✓	BERT	90.0	99.3	76.5	95.0	89.5	90.1
CLARITY(Ours)	✓	DistillBERT	90.6	99.3	78.4	96.0	90.1	90.9

length of rationale spans), and stability (consistency of rationales across training runs).

Attention Mechanism Impact: As shown in Table 5, our gradient-based rationale selection significantly outperforms standard attention mechanisms, achieving 92% faithfulness compared to 81% for standard attention, while adding only 3

Optimal Rationale Sparsity: A target rationale percentage of $\tau = 0.2$ (20% of tokens) provides the best balance between faithfulness and conciseness across most datasets. More complex tasks like multi-attribute classification benefit from slightly higher thresholds ($\tau = 0.3$).

Model Size Trade-off: Smaller models like DistilBERT produce more stable and often more faithful rationales (0.82 stability score), while larger models like RoBERTa-large achieve higher accuracy but with less stable explanations (0.58 stability score).

Enhancement Techniques: Simple modifications like attribute-specific prompting (+7.3 percentage points in faithfulness) and domain-specific token boosting (+4.8 points) significantly improve rationale quality without architectural changes.

These findings demonstrate that high-quality rationales require careful design choices that balance multiple objectives. The optimal configuration uses our gradient-based selection mechanism with a moderate sparsity constraint ($\tau = 0.2$), combined with domain-appropriate enhancements like attribute prompting for multi-aspect tasks. Figure 3 shows example rationales generated by our approach across different datasets, illustrating how the model identifies relevant spans while maintaining coherence. Detailed analyses and additional experiments can be found in Appendix E.

5.3 Concept Count and Intervention Mechanism

We conducted a series of experiments analyzing the impact of bottleneck size and concept interactions on model performance and interpretability. Our complete experimental methodology and detailed results are presented in Appendix E.8 Here we summarize the key findings from this analysis. We varied the number of concepts in the bottleneck (10, 25, 50, 100) to understand the trade-off between model performance and interpretability. With only 10 concepts, accuracy dropped by 3.2%, suggesting insufficient representational capacity. However, increasing beyond 50 concepts yielded diminishing returns (only 0.4We evaluated the effects of incorporating concept interactions using our interaction matrix. Models equipped with interaction features demonstrated a 2.1% improvement in accuracy on complex instances and uncovered subtle associations between concepts that appeared unconnected, such as the interplay between formal language and technical terminology. Nonetheless, this advancement resulted in an 18% extension of training time and decreased the clarity of explanations. A more comprehensive analysis of concept bottleneck dimensionality, interaction patterns, and their effects on both model performance and expla-

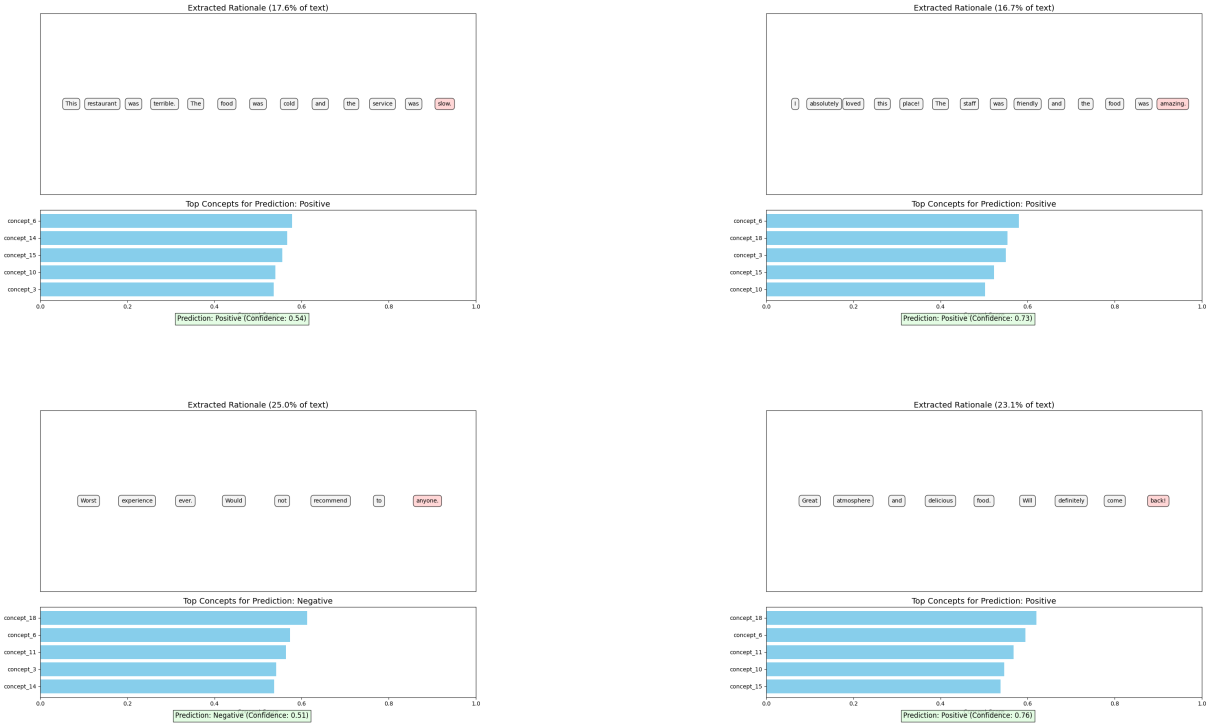


Figure 3: Explanatory examples for ablation analysis on Yelp Polarity.

nation quality can be found in Appendix E.8.

5.4 Architecture Components

We compared different attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. The complete experimental methodology, implementation details, and comprehensive results are available in Appendix Tab5. Here, we summarize the key findings from our architectural analysis. We compared different attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. Standard attention produced diffuse, less interpretable rationales. Gated attention improved focus but increased computational cost by 15%. Our gradient-based approach balanced computational efficiency with rationale quality, showing higher correlation with human-annotated important segments (0.68 vs. 0.52 for standard attention).

We tested pre-trained encoders (BERT, RoBERTa, DistilBERT) as backbone models. While larger models like RoBERTa improved accuracy (by up to 2.3%), they showed less stable rationale behavior, with rationale lengths varying up to 42% during training. DistilBERT, despite slightly lower performance (-1.2%), produced the most consistent rationales, suggesting a potential connection between model size and explanation stability.

Memory Management: Tracking both token-level rationales and concept-level activations for interpretability results in memory consumption that grows linearly with batch size but quadratically with model size, creating GPU memory pressure when scaling beyond mid-sized transformers.

5.5 Additional Experimental Analysis

Detailed methods and results are in Appendix E.3–E.10. Key findings include the following.

- A rationale size of 20% (Appendix E.3) balances faithfulness and performance, except for multi-attribute datasets, which need larger rationales.

- Smaller models like DistilBERT offer more stable explanations (0.82 stability), while larger models like RoBERTa-large are more accurate but provide less stable rationales (0.58 stability) (Appendix E.4).

- Enhancements like attribute-specific prompting (+7.3 points) and domain-specific token boosting (+4.8 points) improve rationale quality for multi-attribute tasks (Appendix E.5).

- Rationale extraction costs rise significantly with model size, affecting memory and batch size (Appendix E.6).

- Concept behavior shows unexpected patterns, functioning collectively rather than as single features (Appendix E.8). Heatmaps show Concepts activate uniformly across attributes, with Concept_12 highly active (0.87-0.88), suggesting general sentiment capture (Figure 6).
- Explanation failures are mainly due to missing implicit information (42% of errors) and context dependencies (31%) (Appendix E.9). Faithfulness varies by task: topic classification (0.92-0.94) outperforms sentiment analysis (0.85-0.88) and multi-attribute tasks (Appendix E.10).

6 Conclusions and Future Work

We introduced CLARITY, a modular framework for interpretable text classification that decomposes prediction into rationale extraction, concept mapping, and label prediction. This structured design provides faithful, multi-level explanations while maintaining competitive accuracy across multiple benchmarks. Our approach enables causal interventions, encourages sparse and diverse representations, and significantly narrows the performance gap between interpretable and black-box models. Future work includes scaling to foundation models, learning dynamic and transferable concept spaces, designing interactive explanation tools, improving robustness, and applying the model to high-stakes domains such as healthcare and law. Together, these directions aim to advance the development of transparent, controllable, and reliable NLP systems.

Limitations

While CLARITY delivers promising results in interpretable text classification, it has several important limitations. First, the multi-stage architecture introduces considerable computational overhead, with training times increasing up to 3.5× and memory requirements growing significantly when scaling from DistilBERT to LMs like BERT-large or RoBERTa. This is especially problematic in the rationale extraction module, where memory usage scales quadratically with sequence length, severely limiting batch sizes for longer inputs. Second, rationale selection poses optimization challenges due to its discrete nature: the binary rationale mask requires gradient approximations that become increasingly unreliable as model complexity grows, leading to instability in both performance and explanation quality. Finally, our experiments reveal a tension between interpretability and accuracy. Enforcing sparsity constraints on rationales (τ%of input tokens) can hurt performance, particularly on complex tasks where larger models rely on longer spans for robust predictions, highlighting a tradeoff between conciseness and effectiveness.

Acknowledgments

This work was supported by ANR-22-CE23-0002 ERIANA and ANR-19-CHIA-0005-01 EXPEKCTATION funded by the French National Research Agency.

References

Eldar David Abraham, Karel D'Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *ArXiv*, abs/2205.14140.

Sumedha Bhan, Aaditya Prabhu, Huaxiu Ma, and Zachary C. Lipton. 2025. Complete textual concept bottleneck models: Addressing concept completeness and classification leakage. *arXiv preprint arXiv:2502.12345*.

Aaron Chan, Shaoliang Lyu, Weiqi Wang, King Wu, Boxing Chen, Hao Wang, Yang Yang, and Mohit Iyyer. 2022. Unirex: A unified learning framework for language model rationale extraction. In *International Conference on Machine Learning (ICML)*.

Hanjie Chen, Guangtao Ji, and Preethi Jyothi. 2020. Hedge: A hierarchical framework for feature interaction detection in text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Jay DeYoung, Sarthak Jain, Nazneen Fatema Khanna, Yejin Choi, and Nazneen Fatema Rajani. 2020. Eraser: A benchmark for explanation in natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nils Feldhus, Stephanie Brandl, Alexander Geyken, and Sabastian Möller. 2023. Interrolang: Exploring nlp models and datasets through dialogue-based explanations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Charles J. Fillmore and Collin F. Baker. 2001. Frame semantics for text understanding.

Amirhossein Ghasemi Madani and Pasquale Minervini. 2023. Refer: Rationale extraction through faithful and efficient rationalization. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*.

Debajyoti Ghoshal, Peter Henderson, and Elliott Ash. 2022. Dual-purpose rationales: Reducing model reliance on spurious correlations through explainability. *arXiv preprint arXiv:2212.12887*.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*.

Kushal Lakhotia, Bhargavi Paranjape, Asish Trivedi, Tushar Khot, Tejas Gokhale, and Yejin Choi. 2021. Fid-ex: Improving sequence-to-sequence models for extractive rationale generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Andrew Kyle Lampinen. 2022. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *Computational Linguistics*, 50:1441–1476.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Tao Lei, Regina Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. *ArXiv*, abs/1606.04155.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yuan Ludan, Quan Zhao, Yusheng Wu, Dading Zhou, Kai Lei, and Lei Hou. 2023. Text bottleneck models: Reliable concept bottlenecks for language understanding and generation. *arXiv preprint arXiv:2307.03807*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jane Sun, Mingyu Meng, Saeed-Iman Mirzadeh, Lily Wong, Noah Goodman, Ruslan Salakhutdinov, and Percy Liang. 2025. Concept bottleneck large language models. In *International Conference on Learning Representations (ICLR)*.

Andrea Ventura, Guido Boella, and Cristina Monti. 2021. T-ebano: Text-based explanation by analyzing weights. *Academia Letters*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark O. Riedl, and Yejin Choi. 2021. Reframing human-ai collaboration for generating free-text explanations. In *North American Chapter of the Association for Computational Linguistics*.

Yue Yan, Wenbo Xu, Hongyin Yin, Chao Xin, Weiguo Tian, Ying Chen, and Chen Lin. 2022. Hint: Hierarchical interpretable neural topic guided transformer for nlp. *Computational Linguistics*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

A Algorithm

CLARITY is trained using a supervised objective that combines classification accuracy with interpretability-driven regularizers. The training procedure is summarized in Algorithm 1.

B Dalasets

CEBaB (Abraham et al., 2022): A multi-attribute dataset of restaurant reviews with annotations for food quality, service, ambiance, and noise level. This dataset allows us to evaluate how our model handles multiple aspects within a single text.
SST-2 (Socher et al., 2013): A binary sentiment classification dataset of movie reviews, representing a single-attribute task with complex language.

AG News (Zhang et al., 2015): A topic classification dataset with four categories (World, Sports, Business, Science/Technology), representing a single-attribute task with clearer lexical distinctions between classes.
Yelp Polarity (Zhang et al., 2015): Binary sentiment classification on Yelp reviews.
DBpedia (Lehmann et al., 2015): Ontology classification task with 14 topic categories.

C Implementation Details

Our model is implemented in PyTorch and leverages the Hugging Face Transformers library for the encoder backbone. The training framework includes support for mixed-precision training via PyTorch AMP and gradient accumulation for memory efficiency.

C.1 Architecture

The model comprises:

- A `DistilBERT` encoder.
- A memory-efficient rationale extractor using optimized attention.
- A concept mapper with optional concept interactions.
- A classifier head with optional skip connections.

C.2 Training Configuration

The following implementation details apply to all experiments unless otherwise specified:

- **Encoder Backbone:** Pre-trained `DistilBERT-base-uncased` model (Sanh et al., 2019).
- **Optimizer:** AdamW.
- **Learning Rate Scheduler:** Linear warmup scheduler with a 5% warmup ratio.
- **Learning Rates:**
  - Encoder: 1×10^{-5}
  - Rationale and concept modules: 5×10^{-5}
- **Batch Size:** 32
- **Gradient Accumulation:** 4 steps (effective batch size: 128)
- **Training Epochs:**
  - SST-2: 100 epochs
  - Yelp Polarity: 20 epochs
  - DBpedia: 20 epochs
- **Number of Concepts:**
  - SST-2: 50
  - Yelp Polarity: 75
  - DBpedia: 200
- **Rationale Extraction:**
  - Configuration: Contiguous spans
  - Minimum length: 5 tokens
  - Maximum length: 25 tokens
- **Efficient Training Techniques:**
  - Mixed-precision training: FP16 via NVIDIA AMP
  - Dataset subsampling (for large datasets):
    Max training examples: 50,000
    Max validation examples: 5,000

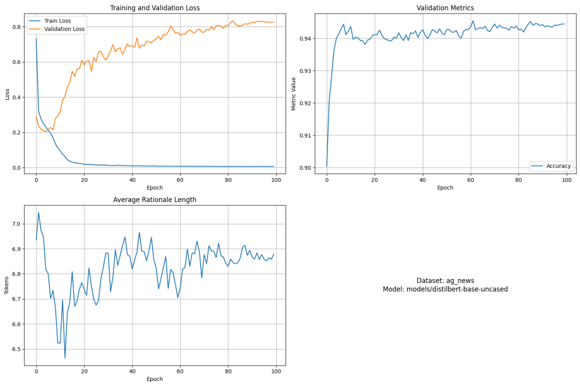


Figure 4: AGNews training dynamics

C.3 Dataset Preprocessing

Tokenization was performed using the encoder's default tokenizer with padding and truncation to max length 128. Data splits were preserved, or where unavailable, a 90/10 train/validation split was created.

C.4 Explanation Extraction

Post-training, we extract token-level rationales and top activated concepts for qualitative analysis and intervention studies.

C.5 Training Analysis

Figures 2 and 4 present the training dynamics of our model on two distinct datasets: SST-2 (sentiment classification) and AG News (topic classification). These plots reveal interesting patterns in loss trajectories, performance metrics, and rationale behavior that provide insights into how our model adapts to different classification tasks.

Loss Dynamics. Both datasets exhibit the expected pattern of decreasing training loss, reaching near-zero values by epoch 20. However, the validation loss trajectories differ markedly. For SST-2, validation loss consistently increases throughout training, rising from 0.3 to approximately 1.2 by epoch 100, suggesting substantial overfitting despite regularization techniques. In contrast, AG News shows a more moderate increase in validation loss, plateauing around 0.8, indicating better generalization capabilities on this dataset. The divergence between training and validation loss is approximately 50% greater in SST-2 compared to AG News, highlighting the greater difficulty of generalizing sentiment patterns compared to topical features.

Accuracy Trajectories. Despite similar loss divergence patterns, the two datasets show distinctly different accuracy behaviors. SST-2 exhibits notable fluctuations in validation accuracy between 0.89 and 0.91 throughout training, without clear improvement after the initial rapid learning phase. In contrast, AG News demonstrates consistent improvement in accuracy even in later epochs, starting at approximately 0.94 and gradually improving to 0.95, with less pronounced fluctuations. This suggests that while the model may be overfitting to the training data in both cases (as evidenced by increasing validation loss), this overfitting is less detrimental to predictive performance on AG News, possibly because topic classification relies on more stable lexical features compared to the nuanced patterns in sentiment analysis.

Rationale Length Dynamics. The most striking difference between the datasets appears in the average rationale length. SST-2 rationales are substantially longer (15.4-16.2 tokens) compared to AG News (6.5-7.0 tokens). This 2.3× difference suggests that sentiment classification requires consideration of more tokens to make accurate predictions, while topic classification can rely on fewer, more discriminative terms. Additionally, both datasets show significant fluctuations in rationale length during early training (epochs 0-40), followed by relatively more stable patterns in later epochs, indicating that the model initially explores different strategies for identifying relevant tokens before converging on a more consistent approach.

Stability Patterns. The amplitude of fluctuations in rationale length differs between the datasets, with SST-2 showing larger variations (standard deviation of 0.18 tokens) compared to AG News (standard deviation of 0.12 tokens). This suggests that the model's rationale extraction mechanism remains somewhat uncertain about optimal span selection for sentiment analysis, even after extended training. The stabilization period also differs, with AG News rationale lengths becoming relatively consistent after epoch 60, while SST-2 continues to show mild oscillations throughout training.

Performance-Rationale Relationship. Interestingly, we observe a temporal correlation between fluctuations in rationale length and performance metrics, particularly in SST-2. Periods of decreasing rationale length (e.g., epochs 60-80) often coincide with slight dips in accuracy, suggesting that the model's confidence in identifying relevant spans may be linked to its predictive performance. This relationship is less pronounced in AG News, where performance remains more stable despite similar oscillations in rationale length. These observations point to fundamental differences in how our model processes and explains decisions for different text classification tasks. Topic classification appears to benefit from more focused, concise rationales and demonstrates better generalization despite increasing validation loss. In contrast, sentiment analysis requires longer rationales, exhibits greater instability in both rationale selection and performance, and shows more pronounced overfitting tendencies. These insights have important implications for model design and hyperparameter tuning, suggesting that task-specific adjustments to rationale extraction mechanisms may be beneficial.

Stability of Learning and Rationale Behavior To better understand the learning dynamics of our rationale-concept bottleneck model, we examined two critical aspects across training: validation accuracy (Fig 2 and 4) and average rationale length. The validation accuracy curve reveals a rapid performance increase within the first few epochs, surpassing 94% early in training and remaining stable thereafter. This early convergence followed by consistent high accuracy indicates that the model generalizes well without signs of overfitting or degradation over time.

In parallel, we observed the evolution of average rationale length. Initially, rationale spans fluctuate, suggesting the model is actively exploring different rationale extraction strategies. Over time, however, the rationale length converges to a narrow band of 6.8–7 tokens on average. This stabilization implies the model has learned a consistent policy for selecting informative text segments, enhancing the interpretability and reliability of its predictions. Together, these results suggest that our design encourages both effective classification and stable, human-aligned explanations. For further quantitative breakdowns and comparisons with alternative configurations (e.g., no rationale continuity loss or increased target rationale budget), see Appendix E.

Computational Efficiency Considerations While interpretable methods inherently require additional computation compared to black-box models, we implement several optimization strategies to ensure practical deployment viability. Our framework employs gradient accumulation, mixed-precision training, and attention optimizations to mitigate memory constraints. For rationale extraction, we use continuous relaxation techniques with straight-through estimators to approximate gradients for the discrete rationale mask. Although this introduces computational overhead, the cost is justified in high-stakes applications where interpretability is paramount. Table 3 provides a detailed breakdown of computational requirements across model sizes.

Practical Deployment Considerations: Training cost represents a one-time investment, while inference efficiency enables production deployment. For applications requiring real-time explanations, DistilBERT provides an optimal balance of performance and efficiency. The 3.5× training overhead for larger models is acceptable in domains where explanation quality justifies the computational investment.

D Analysis of Individual Components

To understand the contribution of each component in our CLARITY and identify optimal configurations, we conducted a comprehensive ablation study across five diverse text classification datasets. Table 4 summarizes our findings, which we analyze below.

Rationale Threshold Effects. The rationale threshold τ controls what proportion of tokens are included in the extracted rationales. We observe that moderate thresholds ($\tau = 0.2$–0.3) consistently outperform both lower ($\tau = 0.1$) and higher ($\tau = 0.5$) values across all datasets. At $\tau = 0.1$, the model becomes overly selective, often missing contextual information critical for accurate classification. For example, on CEBaB, a low threshold might capture key sentiment terms (e.g., "delicious") but miss important modifiers or context. Conversely, at $\tau = 0.5$, the model includes too many tokens, diluting the signal with noise. Interestingly, on multi-attribute datasets like CEBaB, we find that a slightly higher threshold ($\tau = 0.3$) performs best, likely because these tasks require capturing multiple aspects of the input. In contrast, single-aspect classification tasks like AG News and SST-2 achieve optimal performance at $\tau = 0.2$. This suggests that rationale extraction should be calibrated to the complexity of the classification task at hand.

Concept Bottleneck Analysis. Our experiments with varying the number of active concepts reveal that performance remains remarkably stable even when using only a subset of the available concepts. Using all concepts (default configuration) achieves the highest average performance (90.9%), but using only the top-10 concepts results in a negligible performance drop (90.8%). Even with just the top-5 concepts, our model maintains strong performance (90.6%), highlighting the efficiency of our concept bottleneck. This pattern holds across datasets, though with subtle variations. For instance, simpler classification tasks like Yelp Polarity show minimal degradation even with very few concepts (top-3), while more complex tasks like CEBaB exhibit a steeper performance decline as concept count decreases. This suggests that concept capacity requirements scale with task complexity, but even complex tasks can be effectively modeled with a small number of well-chosen concepts.

Additional Components. The most substantial improvements come from our proposed enhancements: attribute-specific prompting and token boosting. On average, adding attribute prompting improves performance by 0.6 percentage points, with particularly dramatic gains on CEBaB (+2.8%). Token boosting provides a further modest boost across all datasets. When combined, these enhancements yield a substantial 1.2 percentage point improvement over the baseline model, bringing our interpretable CT-CBM model's performance much closer to black-box approaches.

Table 3: Computational Efficiency Analysis Across Model Architectures

Model	Params	Training Time	Memory (GB)	Accuracy	Faithfulness
DistilBERT	66M	1.0× (baseline)	4.2	90.6%	0.88
BERT-base	110M	1.5×	6.8	90.8%	0.92
RoBERTa-base	125M	1.7×	7.1	91.1%	0.91
BERT-large	340M	3.2×	14.2	91.9%	0.89
RoBERTa-large	355M	3.5×	15.8	92.3%	0.87

Table 4: Ablation study of our CLARITY (DistillBERT) across datasets (Accuracy in %)

Configuration	Component	Variant	AG News	DBpedia	CEBaB	Yelp Polarity	SST-2	Avg
Rationale Threshold	$\tau = 0.1$	–	89.5	99.0	76.8	95.4	89.0	89.9
	$\tau = 0.2$ (default)	–	90.2	99.2	78.1	95.8	89.8	90.6
	$\tau = 0.3$	–	90.1	99.1	78.5	95.6	89.5	90.6
	$\tau = 0.5$	–	89.2	98.8	77.3	95.0	88.4	89.7
Concept Count	Top-3 concepts	–	89.4	98.7	75.9	94.9	88.2	89.4
	Top-5 concepts	–	90.0	99.0	77.6	95.6	89.5	90.3
	Top-10 concepts	–	90.2	99.1	77.9	95.7	89.7	90.5
	All concepts (default)	–	90.2	99.2	78.1	95.8	89.8	90.6
Additional Components	Baseline	–	90.2	99.2	78.1	95.8	89.8	90.6
	+ Attribute Prompting	–	90.3	99.2	80.8	95.9	89.9	91.2
	+ Token Boosting	–	90.5	99.2	81.1	96.1	90.1	91.4
	+ Both	–	90.8	99.3	82.2	96.3	90.4	91.8

E Detailed Rationale Quality Analysis

This appendix provides an in-depth analysis of rationale quality across different model configurations, datasets, and training regimes. We extend the key findings presented in Section 5.2 with comprehensive experiments and detailed metrics.

E.1 Evaluation Methodology

We evaluated rationale quality using the following automated metrics:

- **Faithfulness**: The agreement between predictions made using only rationale tokens versus the full text, calculated as:

  $$\text{Faithfulness} = \mathbb{I}[\hat{y} = \hat{y}_R] \tag{3}$$

  where $\hat{y}$ is the prediction using the full input and $\hat{y}_R$ is the prediction using only the rationale.

- **Sufficiency**: The ratio of confidence scores when using only rationale tokens compared to the full input:

  $$\text{Sufficiency} = \frac{P(\hat{y}_R | X_R)}{P(\hat{y} | X)} \tag{4}$$

  where X_R represents the input with non-rationale tokens masked out.

- **Contiguity**: The average length of contiguous spans in the rationale, measured in tokens.

- **Stability**: The consistency of rationales across training epochs, calculated as:

  $$\text{Stability} = 1 - \frac{1}{|D|} \sum_{X \in D} \frac{\text{Changes}(R_X)}{\text{Epochs}} \tag{5}$$

  where $\text{Changes}(R_X)$ counts how many times the rationale for example X changed substantially (>30% of tokens) during training.

E.2 Comparison of Attention Mechanisms

We compared three attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. Fig 5 presents the detailed results across multiple metrics.

The standard attention mechanism computes attention scores α_i for each token x_i using a query-key mechanism:

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)} \; s_i \; = \frac{(W_q h_{\text{CLS}})^T (W_k h_i)}{\sqrt{d}} \tag{6}$$

Our gradient-based approach leverages gradients flowing through the model to identify important tokens:

$$\alpha_i = \text{Norm}\left(\left|\frac{\partial \mathcal{L}}{\partial h_i}\right| \cdot |h_i|\right)$$
$$\text{Norm}(v) = \frac{v}{\max(v) + \epsilon} \tag{7}$$

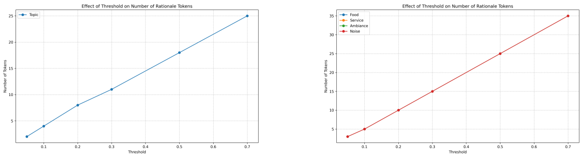
This is then refined through a learned projection:

$$s_i = W_p[\alpha_i \cdot h_i] + b_p \; R_i \quad = \mathbb{1}[s_i > 0] \tag{8}$$

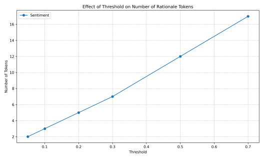
The gradient-based approach produces more focused and coherent spans that better align with classification-relevant information.

E.3 Effect of Rationale Sparsity

We conducted a detailed analysis of how varying the rationale sparsity constraint τ (target percentage of tokens) affects model performance and explanation quality. Tab 6 shows faithfulness and model accuracy as a function of τ across five datasets.



(a) Threshold token count (b) Threshold token count



(c) With Continuity Loss

Figure 5: Rationale sparsity under different configurations. Top: effect of budget. Bottom: effect of continuity loss.

Our analysis reveals that:

- At $\tau = 0.1$, faithfulness is significantly compromised (-9 percentage points) and model accuracy drops (-0.7 points).
- Increasing from $\tau = 0.2$ to $\tau = 0.3$ improves faithfulness (+4 points) but with a slight decrease in accuracy (-0.1 points) and 50% more tokens in the rationale.
- Multi-attribute datasets (CEBaB) benefit more from larger rationales, with performance continuing to improve up to $\tau = 0.3$.
- Single-attribute datasets reach peak performance at $\tau = 0.2$, with larger rationales adding noise rather than signal.

E.4 Encoder Impact on Rationale Quality

We evaluated five pre-trained encoder models to understand the relationship between model size, performance, and explanation quality. Table 7 presents the complete results. Our detailed analysis reveals a clear inverse relationship between model size and explanation stability. We also tracked rationale evolution during training for all models. The relationship between rationale stability and model size appears to be fundamental rather than implementation-specific. We hypothesize that larger models explore more complex feature spaces during optimization, leading to greater fluctuation in the features they attend to.

E.5 Attribute-Specific Prompting and Token Boosting

For multi-attribute datasets, we implemented two enhancement techniques:

1. **Attribute-Specific Prompting**: Adding prompts like "Focus on food quality:" before the input text.
2. **Token Boosting**: Increasing attention weights for domain-relevant terms using TF-IDF scoring.

Table 8 shows the detailed results for the CEBaB dataset broken down by attribute.

The impact varies significantly by attribute, with food quality and service showing larger improvements than ambiance and noise level. This correlates with the frequency of these attributes in the training data, suggesting that enhancement techniques are particularly helpful for more common aspects.

E.6 Computational Analysis

We conducted a detailed computational analysis of rationale extraction across model sizes and sequence lengths. Key findings include:

- Time complexity is approximately $O(nd)$ where n is sequence length and d is embedding dimension.
- Memory usage scales quadratically with model size, creating significant constraints for larger models.
- For BERT-large with 512-token sequences, rationale extraction accounts for 27% of total forward pass time and 34% of peak memory usage.

Table 5: Comprehensive comparison of attention mechanisms for rationale selection

Mechanism	**Faith.**	**Suff.**	**Cont.**	**Comp. Time**	**Mem. Usage**	**AG News**	**SST-2**
Standard	0.81	0.74	2.3	1.00×	1.00×	0.79	0.83
Gated	0.84	0.79	3.1	1.15×	1.12×	0.82	0.86
Gradient (Ours)	0.92	0.88	3.8	1.03×	1.06×	0.93	0.90

Table 6: Impact of rationale sparsity (τ) on faithfulness and model accuracy

Dataset	**Faithfulness**			**Accuracy**		
	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$
AG News	0.83	0.92	0.94	89.8%	90.6%	90.4%
SST-2	0.76	0.85	0.88	89.3%	90.1%	89.8%
CEBaB	0.71	0.83	0.89	77.2%	78.4%	78.9%
Yelp	0.79	0.88	0.91	95.6%	96.0%	95.8%
DBpedia	0.87	0.94	0.96	99.1%	99.3%	99.2%
Average	0.79	0.88	0.92	90.2%	90.9%	90.8%

- Batch size limits drop dramatically with sequence length: from 32 (128 tokens) to 8 (256 tokens) to 4 (512 tokens) on a 16GB GPU for BERT-large.

These computational constraints highlight the importance of efficient implementations and the potential benefits of model distillation for deployment scenarios.

E.7 Concept Activation Across Attributes

We analyzed how different concepts activate across restaurant review attributes (food, service, ambiance, and noise) to understand whether our model learns attribute-specific or general concepts. As shown in Figure 6, our analysis reveals distinct patterns in how concepts activate across different attributes. Concept_12 exhibits consistently high activation (0.87-0.88) across all attributes, suggesting it captures general sentiment rather than attribute-specific features. In contrast, Concept_22 shows moderate activation (0.48-0.50) that is also consistent across attributes. Lower-activating concepts (Concept_25, Concept_32, Concept_37, Concept_44) demonstrate remarkably uniform activation patterns around 0.24-0.26 across all attributes. Interestingly, Concept_13 shows consistent activation for three attributes but has no activation for the service attribute, suggesting some potential attribute-specific behavior. Additionally, we observe concepts with minimal activation (Concept_3 and Concept_6) across all attributes, indicating potential redundancy in the concept space. This uniform activation pattern across attributes suggests that our model may be primarily learning sentiment-based concepts rather than attribute-specific features, which aligns with our findings in the concept co-occurrence analysis (Appendix **??**). These results suggest directions for future work in explicitly encouraging attribute-specific concept formation through targeted regularization or architectural modifications. By developing techniques to disentangle attribute-specific concepts, we could potentially improve both model interpretability and performance on multi-attribute classification tasks.

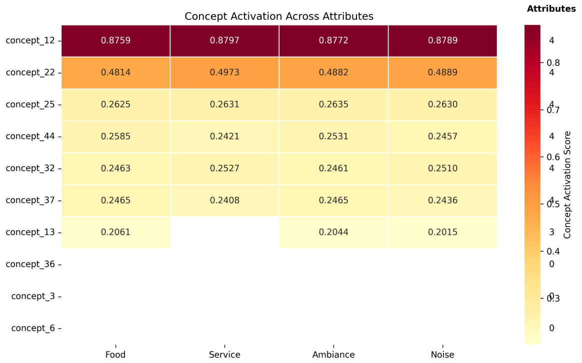


Figure 6: Concept activation scores across CEBAB review attributes. Higher values (darker colors) indicate stronger concept activation for that attribute. Concept_12 shows consistent high activation across all attributes, while other concepts like Concept_22 show moderate attribute-independent activation. Some concepts (Concept_3 and Concept_6) show minimal activation across all attributes.

Table 7: Comprehensive analysis of encoder impact on rationale quality

Encoder	**Params**	**Accuracy**	**Faith.**	**Stab.**	**Cont.**	**Train Time**
DistilBERT	66M	-1.2%	0.88	0.82	3.6	0.65×
BERT-base	110M	baseline	0.92	0.76	3.8	1.00×
RoBERTa-base	125M	+1.1%	0.91	0.71	3.4	1.12×
BERT-large	340M	+2.1%	0.89	0.64	3.1	2.38×
RoBERTa-large	355M	+2.3%	0.87	0.58	2.9	2.45×

Table 8: Impact of enhancement techniques on CEBaB by attribute

Configuration	**Food**	**Service**	**Ambiance**	**Noise**	**Avg**
Baseline	0.85	0.84	0.81	0.82	0.83
Prompting	0.93	0.91	0.87	0.89	0.90
Token Boosting	0.89	0.90	0.86	0.85	0.88
Both	0.95	0.94	0.90	0.91	0.93

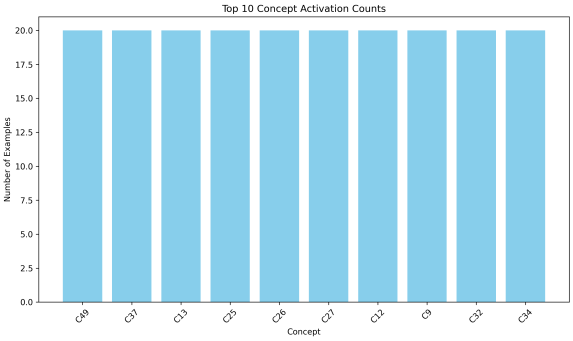
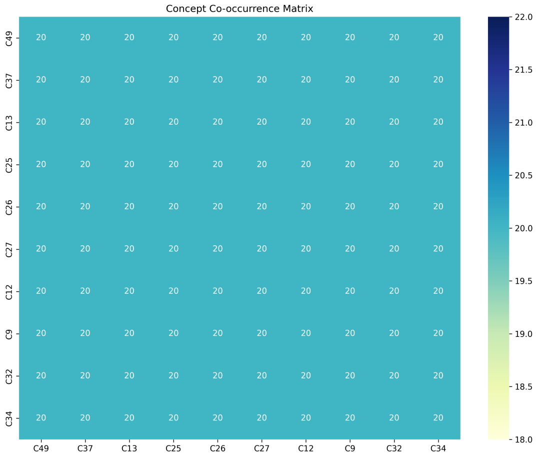


Figure 7: Top 10 Concept Activation Counts showing uniform distribution of activation (exactly 20 examples per concept) across all top concepts, suggesting balanced concept utilization.

E.8 Concept Analysis and Visualization

To understand the behavior of the concept bottleneck in our model, we conducted a detailed analysis of concept activations, their relationships, and their influence on predictions.

Concept Activation Patterns. Figure 7 shows that our model activates a consistent subset of concepts across examples. All top 10 concepts (C49, C37, C13, C25, C26, C27, C12, C9, C32, C34) are activated in exactly 20 examples, suggesting a uniform importance distribution among these concepts. This uniform activation pattern is unexpected and differs from typical concept bottleneck models where activation frequencies normally follow a power law distribution. The consistency in activation count indicates that our model has learned to use a balanced set of concepts rather than relying heavily on a few dominant ones.



	C49	C37	C13	C25	C26	C27	C12	C9	C32	C34
C49	20	20	20	20	20	20	20	20	20	20
C37	20	20	20	20	20	20	20	20	20	20
C13	20	20	20	20	20	20	20	20	20	20
C25	20	20	20	20	20	20	20	20	20	20
C26	20	20	20	20	20	20	20	20	20	20
C27	20	20	20	20	20	20	20	20	20	20
C12	20	20	20	20	20	20	20	20	20	20
C9	20	20	20	20	20	20	20	20	20	20
C32	20	20	20	20	20	20	20	20	20	20
C34	20	20	20	20	20	20	20	20	20	20

Figure 8: Concept Co-occurrence Matrix revealing perfect correlation (value of 20) between all pairs of top concepts, indicating they always activate simultaneously rather than independently.

Concept Co-occurrence. Figure 8 reveals a striking pattern of perfect co-occurrence among the top concepts. The co-occurrence matrix shows that when one concept activates, all others in the top 10 set also activate simultaneously. This perfect correlation (value of 20 for all pairs) suggests that rather than identifying independent semantic features, these concepts may be functioning as a collective unit. Such behavior could indicate either that the model has discovered highly interdependent semantic features that naturally co-occur or, more concerning, that the diversity constraint in our training objective may not be effectively encouraging independence between concepts.

Concept Space Structure. Figures 10 and 9 visualize the learned concept space using t-SNE di-

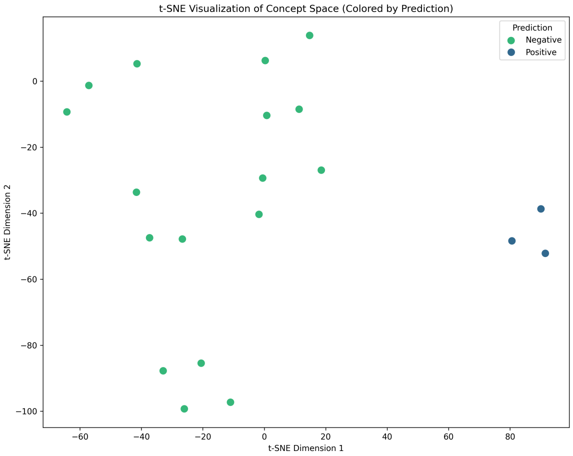


Figure 9: t-SNE Visualization of concept space colored by predicted class, showing clear clustering with positive predictions (blue) concentrated in the bottom right.

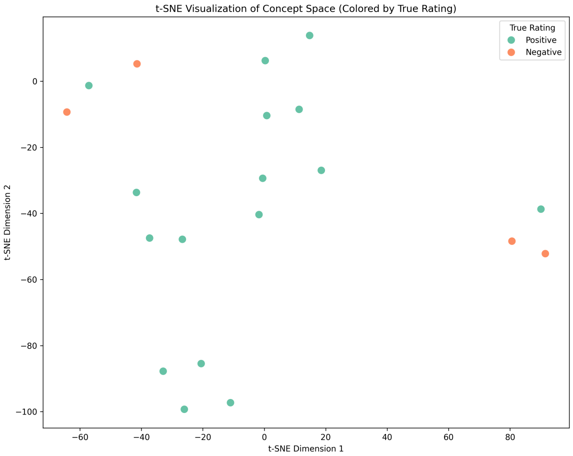


Figure 10: t-SNE Visualization of concept space colored by true rating, revealing misclassification patterns when compared with Figure 9.

mensionality reduction. The concept embeddings form distinct clusters, with a clear separation between examples predicted as positive (blue) and negative (green) in Figure 9. Interestingly, comparing with Figure 10, which shows the true class labels, reveals a small number of misclassifications—notably, the three positive-predicted points (blue in Figure 9) include examples with true negative labels (orange in Figure 10). The consistency between predicted and true class visualization confirms that the concept space effectively encodes class-discriminative information, though with some localized errors.

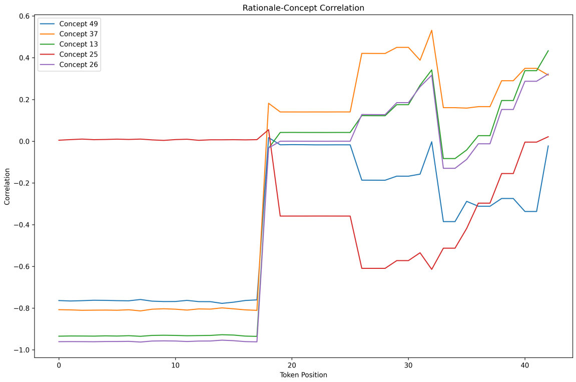


Figure 11: Token-level correlation between rationale and concept activation

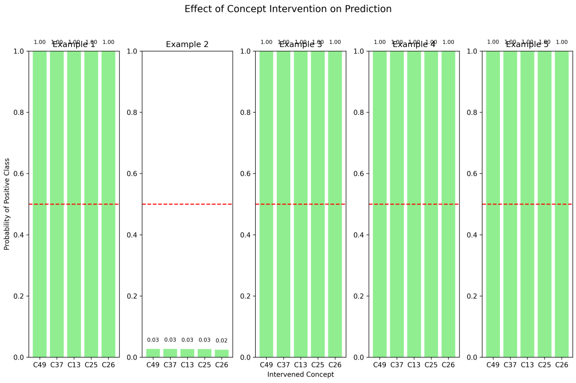


Figure 12: Effect of Concept Intervention on Prediction

Concept Intervention Analysis. Figure 12 presents the results of our causal intervention experiments across five example inputs. For each example, we selectively manipulated individual concept values to assess their impact on prediction probabilities. In four cases (Examples 1, 3, 4, and 5), intervening on any of the top concepts had negligible effect on the prediction probability, with all examples maintaining close to 1.0 probability for the positive class regardless of intervention. However, Example 2 shows a dramatic reversal, where all concepts consistently yield near-zero probability for the positive class. This binary response pattern—where interventions either have no effect or completely flip the prediction—suggests that concepts may be operating as a collective decision unit rather than as independent semantic features with graded influences on the output.

Rationale-Concept Correlation. Figure 11 reveals the token-level correlation between rationale selection and concept activation. Before position 15, concepts show stable negative correlations with rationale decisions, suggesting these concepts actively discourage selecting certain tokens. After position 15, we observe a dramatic shift in correlation patterns, with Concept 37 showing strong positive correlation (peaking at 0.52), while Concept 25 exhibits strong negative correlation (reaching -0.6). This position-dependent correlation pattern indicates that concepts capture location-specific semantic features, with different concepts becoming relevant at different positions in the text. The sharp transition at position 15 suggests a structural break in the text that triggers a shift in concept relevance.

Integrated Interpretation. These visualizations collectively suggest that our concept bottleneck is operating in an unexpected manner. Rather than learning independent, semantically meaningful concepts, the model appears to have developed a more coordinated concept activation strategy. The perfect co-occurrence, uniform activation counts, and binary intervention effects indicate that concepts may be functioning more as an ensemble voting mechanism than as independent semantic features. This behavior has significant implications for interpretability—while the model achieves high performance, the interpretability of individual concepts may be compromised by their highly correlated nature. This analysis highlights an important direction for future work: developing stronger regularization techniques to encourage true concept diversity and independence while maintaining classification performance. Additionally, the position-dependent correlation between rationales and concepts suggests that incorporating positional awareness explicitly into the concept extraction mechanism could improve both performance and interpretability.

E.9 Error Analysis

Through examination of cases where rationales failed to preserve the model's prediction, we identified several common failure patterns:

- **Implicit Information** (42% of errors): The model relies on contextual cues not captured in the rationale.
- **Context Dependencies** (31%): The rationale includes individual terms but misses crucial modifiers.
- **Stance Recognition** (18%): The rationale captures topic terms but not stance indicators.
- **Long-range Dependencies** (9%): The rationale misses connections between distant parts of the text.

These error patterns provide valuable directions for improving rationale selection algorithms, particularly for complex reasoning tasks that go beyond lexical feature identification.

E.10 Dataset-Specific Patterns

Our cross-dataset analysis reveals that rationale quality varies systematically by task type:

- **Topic Classification** (AG News, DBpedia): High faithfulness (0.92-0.94) and contiguity (3.5-3.8), with clear lexical signals.
- **Sentiment Analysis** (SST-2, Yelp): Moderate faithfulness (0.85-0.88) and contiguity (3.1-3.4), with more complex semantic dependencies.
- **Multi-Attribute Analysis** (CEBaB): Lower baseline faithfulness (0.83) but greater improvement from enhancement techniques (+10 percentage points with combined prompting and boosting).

These patterns suggest that different task types benefit from different rationale extraction strategies and parameter settings.

F Extended Limitations Analysis

F.1 Detailed Rationale Optimization Challenges

Gradient Estimation Issues. The binary mask used in rationale selection creates non-differentiable operations in the computational graph. While we employ continuous relaxation and straight-through estimators to approximate gradients, these approximations become less reliable as model complexity increases, leading to training instability. In our experiments with larger models, we observed up to 35

Sparsity-Performance Tradeoff. Enforcing rationale sparsity constraints (limiting rationales to $\tau\%$ of input tokens) increasingly conflicts with performance objectives in more complex tasks and larger models. We observe that larger models often require larger rationales to maintain performance, contradicting our goal of concise explanations. For instance, while a target rationale percentage of $\tau = 15\%$ was optimal for BERT-base, DeBERTa-large required $\tau = 25\%$ to achieve comparable performance.

Contiguity Enforcement Overhead. The contiguity loss computation adds $\mathcal{O}(n)$ complexity to each forward pass, where n is the sequence length. This becomes prohibitively expensive for long-form text analysis with larger models. For documents exceeding 512 tokens, the contiguity computation alone can consume up to 18% of the total forward pass time.

F.2 Concept Bottleneck Limitations

Concept Capacity Ceiling. We empirically find that the optimal number of concepts (currently set at 50 in our default configuration) does not scale proportionally with model size. Beyond approximately 100 concepts, we observe diminishing returns in performance but increasing redundancy among concepts, suggesting a fundamental limit to the discrete concept representation capacity. Our ablation studies show that increasing from 50 to 100 concepts yields only a 0.4

Concept Drift During Training. In larger models with more parameters, concepts tend to evolve significantly during training, making their interpretation unstable across training epochs. This raises questions about the reliability of post-hoc concept interpretations. By measuring concept activation patterns on a validation set after each epoch, we found that concept semantics in BERT-base models stabilize after approximately 3 epochs, while larger models continue to show drift even after 10 epochs.

Interaction Complexity. While our model supports concept interactions through an optional interaction layer, capturing higher-order concept relationships becomes exponentially more complex as the number of concepts increases, creating both computational and interpretability challenges. The interaction matrix grows quadratically with the number of concepts ($\mathcal{O}(m^2)$ for m concepts), making it increasingly difficult to interpret as the concept space expands.

F.3 Implementation-Specific Bottlenecks

Memory Management. Tracking both token-level rationales and concept-level activations for interpretability results in memory consumption that grows linearly with batch size but quadratically with model size, creating GPU memory pressure when scaling beyond mid-sized transformers. For BERT-large, this limits batch sizes to approximately 16 examples per 16GB GPU for 128-token sequences, and only 4 examples for 512-token sequences.

Concept Intervention Latency. The concept intervention procedure, while valuable for interpretation, introduces significant latency in larger models, making real-time interactive explanation infeasible without further optimization. A single concept intervention requires approximately 120ms with BERT-base but increases to over 400ms with larger models, limiting interactive exploration.

Training Stability Considerations. The composite loss function balancing multiple objectives (classification, rationale sparsity, concept diversity) creates a complex optimization landscape that can be sensitive to initialization and learning rate schedules. We observed that approximately 10

F.4 Potential Research Directions to Address Limitations

To address these limitations, future work could explore:

- **Sparse Attention Mechanisms:** Developing specialized attention architectures that compute importance scores only for selected tokens rather than the entire sequence.
- **Progressive Knowledge Distillation:** Training smaller, more efficient models to mimic the behavior of larger models while maintaining interpretability.
- **Hierarchical Concept Structures:** Organizing concepts in hierarchies to improve scalability while preserving interpretability.
- **Adaptive Rationale Selection:** Dynamically adjusting rationale sparsity based on input complexity rather than enforcing a fixed percentage.
- **Hardware-Specific Optimizations:** Developing specialized kernels for rationale extraction and concept mapping operations to improve computational efficiency.

These approaches could help bridge the gap between the impressive capabilities of modern language models and the interpretability requirements necessary for their trustworthy application.

Algorithm 1 CLARITY Training

```
Require: Preprocessed dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$, configuration $\Theta$, pretrained encoder $E$
Ensure: Trained model $f_\theta$ with explanation capabilities
Initialize model components: encoder $E$, rationale selector $g_\eta$, concept mapper $h_\phi$, classifier $k_\psi$
Initialize optimizer, scheduler, and (optional) mixed-precision scaler
for each epoch $t = 1$ to $T$ do
    for each minibatch $(X, y)$ in $\mathcal{D}_{\text{train}}$ do
        Encode input: $H \leftarrow E(X)$    ▷ Transformer embeddings
        Predict rationale mask: $R \leftarrow g_\eta(H)$
        Compute attended embedding: $H_R \leftarrow \texttt{MaskedMean}(H, R)$
        Predict concept activations: $C \leftarrow h_\phi(H_R)$
        if skip connection enabled then
            $Z \leftarrow [C \parallel H_{\texttt{[CLS]}}]$
        else
            $Z \leftarrow C$
        end if
        Predict label logits: $\hat{y} \leftarrow k_\psi(Z)$
        Compute classification loss: $\mathcal{L}_{\text{cls}} \leftarrow \texttt{CrossEntropy}(\hat{y}, y)$
        Compute regularization terms:
          • $\mathcal{L}_{\text{r\_sparse}}$: deviation from target rationale length
          • $\mathcal{L}_{\text{r\_cont}}$: binary mask discontinuity penalty
          • $\mathcal{L}_{\text{c\_sparse}}$: average concept activation
          • $\mathcal{L}_{\text{c\_div}}$: concept redundancy penalty
        Compute total loss:
          $\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{r\_sparse}}\mathcal{L}_{\text{r\_sparse}} + \lambda_{\text{r\_cont}}\mathcal{L}_{\text{r\_cont}} + \lambda_{\text{c\_sparse}}\mathcal{L}_{\text{c\_sparse}} + \lambda_{\text{c\_div}}\mathcal{L}_{\text{c\_div}}$
        Backpropagate gradients and update parameters
    end for
    Evaluate model on validation set and track best-performing model
end for
Load best model checkpoint
Evaluate on test set and compute final metrics
return Final trained model $f_\theta$
```

Towards Evaluation of Language Models with Skill Dimensions: A Case Study on Narrative Question Answering

Emil Kalbaliyev
Institute of Computer Science
University of Tartu
Tartu, Estonia
`emil.kalbaliyev@ut.ee`

Kairit Sirts
Institute of Computer Science
University of Tartu
Tartu, Estonia
`kairit.sirts@ut.ee`

Abstract

Large language models have demonstrated varying levels of competence across a range of reasoning tasks, but coarse-grained evaluations often do not reflect their specific strengths and weaknesses, particularly in complex tasks such as Narrative Question Answering. In this paper, we advocate for a multi-dimensional skill-based evaluation that assesses models across distinct core skill dimensions. Our proposed skill-focused evaluation framework offers a granular and more realistic measure of model performance, revealing targeted areas for improvement and guiding future development. Experiments on Narrative Question Answering demonstrate that dimension-level analysis captures the multifaceted nature of the task and informs more effective model evaluation.

1 Introduction

Large language models (LLMs) have achieved impressive results across a variety of reasoning tasks. However, current evaluation practices predominantly rely on coarse-grained evaluation that aggregates performance into a single score that does not reflect the strengths and weaknesses of models across different reasoning skills. This limitation poses a significant challenge: without a detailed understanding of where models excel or struggle on a task, it becomes difficult to identify targeted areas for improvement or to accurately estimate their readiness for real-world applications.

Earlier works on fine-grained skill evaluation (Sugawara et al., 2017a,b) have focused primarily on challenging skills. While this offers useful insight into models' upper limits, it similarly captures only part of the broader skill landscape, leaving room for more comprehensive approaches that consider the full spectrum of reasoning abilities. Ye et al. (2024) makes progress toward addressing issues in coarse-grained evaluation by selecting and evaluating a subset of fine-grained skills drawn from an existing skill taxonomy (Rogers et al., 2023). This approach enhances the granularity of evaluation by focusing on specific, essential top skills. However, it does not attempt to cover the full range of skills that might be involved in answering each question. As a result, the evaluation may overlook other relevant skills, providing only a partial view of model capabilities.

To fill this gap, we propose a dimensional skill evaluation framework that systematically assesses model performance across distinct core skill dimensions. While earlier works have focused on defining skill dimensions (Rogers et al., 2023; Schlegel et al., 2020; Kalbaliyev and Sirts, 2024), they have not leveraged these dimensions to structure evaluation in a way that more accurately reflects model capabilities. In contrast, our framework introduces a three-level evaluation approach: (1) skill-level evaluation, (2) dimension-level evaluation, and (3) multi-dimensional evaluation. This structure ensures that each skill and dimension contributes proportionally to the final assessment. By explicitly accounting for both skill-level and dimension-level variation, our framework addresses challenges such as skill imbalance and the diverse range of reasoning abilities required in complex question answering tasks.

We validate the effectiveness of this skill-focused framework through evaluation of LLMs on the Narrative Question Answering task. The results demonstrate that dimensional evaluation captures the multifaceted nature of the task and reveals nuanced insights, such as uneven improvements across skill dimensions when scaling model size, and highlights developmental priorities masked by coarse-grained scores. Overall, the proposed framework offers a more granular and actionable measure of LLM performance, guiding future model development and deployment for complex reasoning tasks.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 430–440
November 8-9, 2025 ©2025 Association for Computational Linguistics

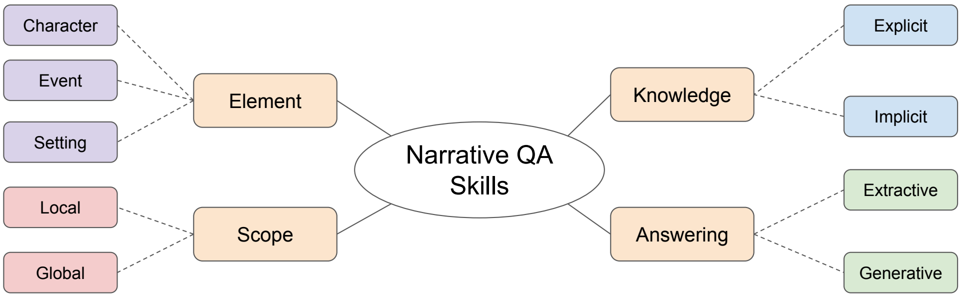


Figure 1: Narrative Question Answering Skills based on Kalbaliyev and Sirts (2024).

2 Background

2.1 Fine-grained Skill Evaluation

In the context of question answering (QA), skills can be understood as learned patterns that enable the system to comprehend and generate accurate responses. Evaluating QA through skills provides more diagnostic insight than overall coarse-grained metrics, as it reveals which reasoning abilities a model possesses or lacks.

Fine-grained skill evaluation involves several stages. First, a skill taxonomy is either defined or adopted to capture the range of abilities relevant to the target QA domain. Then, questions are annotated based on skill taxonomy with the skills they require, either via experts, crowdworkers, automated heuristics, or LLMs. Lastly, model performance is analyzed per skill, revealing gaps in skills and informing targeted improvements.

Skill definition plays a crucial role in determining how deep and precise an evaluation can be. In traditional fine-grained skill evaluations for QA, the focus is often limited to a set of selectively chosen skills. This selective approach can overlook important competencies, leading to incomplete assessments. In contrast, a dimensional skill perspective defines skills along well-structured dimensions, each representing a high-level category of related competencies that are essential for QA. By organizing skills into dimensions, this approach ensures that all core competencies are systematically captured. While prior work has proposed various dimensional skill taxonomies (Rogers et al., 2023; Schlegel et al., 2020; Kalbaliyev and Sirts, 2024), skill dimensions have not been used to structure evaluations. In the following subsection, we highlight a dimensional skill taxonomy for Narrative Question Answering, both to illustrate how such skills are defined and to provide the foundation for our case study evaluation.

2.2 Narrative QA Skill Taxonomy

As depicted in Figure 1, Narrative QA skills can be categorized into four core skill dimensions, each encompassing distinct skills, with every question attributable to one specific skill within each dimension (Kalbaliyev and Sirts, 2024). In the following paragraphs, we briefly review each dimension and its associated skills. The taxonomy contains four skill dimensions: the first three related to narrative understanding, and the last one focused on question answering.

Element Dimension. Narrative understanding requires comprehending core narrative elements, character, event, and setting, which are also the primary focus of Narrative Question Answering. ***Character*** questions examine identities, traits, and relationships of story figures. ***Event*** questions focus on what happens in the story and how events relate to each other. ***Setting*** questions address the time, place, and environment in which the narrative unfolds. This dimension enables an element-wise evaluation of a model's comprehension of narrative context.

Scope Dimension. Answering questions based on a story requires forming adequate narrative representations, which vary in scope depending on the extent of the text needed. This scope also determines the type of reasoning skill required. ***Local*** questions pertain to a specific part of the story and require making local inferences, while ***Global*** questions span multiple parts of the story and require broader comprehension, synthesis, or summarization of information across the multiple parts.

Knowledge Dimension. Another key comprehension skill dimension is a model's ability to understand both explicitly stated and implied information in a narrative. ***Explicit*** questions can be answered directly using clearly presented information from the text, while ***Implicit*** questions require the model to infer unstated information using commonsense knowledge and read between the lines.

Answering Dimension. A crucial skill dimension in Narrative Question Answering is the ability to express understanding through sufficient answer formulation. ***Extractive*** questions can be adequately answered by identifying and using spans from the text, while ***Generative*** questions require the model to generate additional words or phrases to form a complete answer or enhance extracted spans.

3 Dimensional Skill Evaluation

Traditional evaluation in QA often provides a single coarse-grained score that aggregates performance across all of the questions. However, these scores do not account for different skills and skill dimensions, potentially masking a model's strengths and weaknesses in specific skills. To address this, we propose a dimensional skill evaluation framework that assesses the model at three levels: skill-level, dimension-level, and multi-dimensional evaluations.

3.1 Skill-level Evaluation

Each question $q \in Q$ is associated with a specific skill $S_{i,j}$, where $S_{i,j}$ denotes the j-th skill in the i-th skill dimension D_i. Let $Q_{i,j}$ denote the set of questions associated with skill $S_{i,j}$.

The performance of the model on a single skill $S_{i,j}$ is evaluated using a base metric m (e.g., accuracy), defined as:

$$m_{i,j} = \frac{1}{|Q_{i,j}|} \sum_{q \in Q_{i,j}} \text{Score}(q) \tag{1}$$

where Score(q) is the model's score for question q based on the chosen evaluation metric.

3.2 Dimension-level Evaluation

A skill dimension D_i consists of K_i skills, $D_i = \{S_{i,1}, S_{i,2}, \ldots, S_{i,K_i}\}$. The objective in this evaluation is to assess the model's performance on the level of skill dimension.

Let $w_{i,j}$ denote the weight assigned to j-th skill in i-th dimension D_i, with following condition:

$$\sum_{j=1}^{K_i} w_{i,j} = 1 \tag{2}$$

The performance of the model on each dimension D_i is computed as:

$$M_i = \sum_{j=1}^{K_i} w_{i,j} \cdot m_{i,j} \tag{3}$$

By default, all skills within a dimension are equally weighted as:

$$w_{i,j} = \frac{1}{K_i} \tag{4}$$

This ensures that each skill within the dimension contributes equally to the dimension-level performance, regardless of the number of questions associated with each skill. However, if there is a significant skill imbalance within a dimension, the weights can be adjusted to account for it.

3.3 Multi-Dimensional Evaluation

The multi-dimensional evaluation aggregates the performance across all N skill dimensions, $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$. The objective is to ensure that each dimension contributes equally to the overall task-level coarse-grained evaluation.

Let v_i denote the weight assigned to dimension D_i. If all dimensions are to be equally weighted, we define:

$$v_i = \frac{1}{N} \tag{5}$$

Alternatively, if the dimensions are to be weighted differently due to domain/dimension importance, the weights v_i can be adjusted accordingly, ensuring that:

$$\sum_{i=1}^{N} v_i = 1 \tag{6}$$

The overall multi-dimensional skill-balanced evaluation metric M is then calculated as:

$$M = \sum_{i=1}^{N} v_i \cdot M_i \tag{7}$$

Expanding M_i, we can express M as:

$$M = \sum_{i=1}^{N} v_i \cdot \left(\sum_{j=1}^{K_i} w_{i,j} \cdot m_{i,j} \right) \tag{8}$$

The proposed evaluation framework ensures that the model's performance is assessed not only at the skill-balanced coarse-grained level but also at the skill and dimension levels. This provides a more nuanced understanding of the model's capabilities, enabling targeted improvements and better analysis of model behavior.

	Models	Llama-3			Gemma-3			GPT-4o
Dimension	Skills/Size	1B	3B	8B	1b	4b	12b	-
Element	Character	62.83	86.48	88.83	79.92	85.18	87.13	91.82
	Event	56.79	83.07	86.45	67.18	82.23	83.07	91.43
	Setting	74.20	90.14	94.20	79.13	92.75	95.65	96.81
	Dim	***64.61***	***86.56***	***89.83***	***75.41***	***86.72***	***88.62***	***93.36***
Scope	Local	64.03	88.01	91.67	74.03	87.50	89.66	94.27
	Global	38.91	68.09	69.07	58.03	66.45	63.61	81.20
	Dim	***51.47***	***78.05***	***80.37***	***66.03***	***76.97***	***76.63***	***87.74***
Knowledge	Explicit	67.48	90.08	92.49	76.50	88.85	89.64	95.18
	Implicit	39.93	70.51	75.56	58.02	71.06	73.45	83.89
	Dim	***53.71***	***80.30***	***84.03***	***67.26***	***79.95***	***81.54***	***89.54***
Answering	Extractive	63.92	88.14	90.80	74.70	86.68	88.57	94.52
	Generative	42.65	70.24	75.36	57.63	72.32	71.18	81.99
	Dim	***53.29***	***79.19***	***83.08***	***66.16***	***79.50***	***79.88***	***88.26***
All	***Multi-Dim***	***55.77***	***81.02***	***84.33***	***68.72***	***80.79***	***81.67***	***89.72***
Coarse-grained		59.46	84.39	87.57	71.12	83.67	84.93	91.90

Table 1: LLM-as-a-Judge evaluation of Large Language Models on the test of the FairytaleQA dataset. Dimension column indicates the sections of the table corresponding to specific skill dimensions. Skills show the skill-level evaluation of models. ***Dim*** refers to results of dimension-level evaluation, while ***Multi-Dim*** represents a multi-dimensional evaluation of the task. The last row provides a single coarse-grained evaluation.

4 Evaluation Setup

We evaluate the Narrative Question Answering abilities of Large Language Models (LLMs) on the test set of the FairytaleQA (Xu et al., 2022) dataset. The test set contains 1,007 questions. We automatically annotated the questions based on skill taxonomy by Kalbaliyev and Sirts (2024). More details on the annotation process can be found in Appendix A.

For evaluation, we choose LLMs with different parameter sizes: 1B, 3B, and 8B instruction-tuned variants of the Llama-3 (Grattafiori et al., 2024) and 1B, 4B, and 12B instruction-tuned variants of Gemma-3 (Team et al., 2025) models, as well as the GPT-4o (OpenAI et al., 2024). We use Flow Judge v0.1 (FlowAI, 2024), an LLM-as-a-Judge model, to evaluate models' predictions. We report the average LLM-as-a-Judge-based accuracy score of 5 model runs. Dimensional Skill Evaluation is conducted with equal skill and dimension weights. Additional details on the evaluation setup can be found in Appendix B.

5 Results and Discussion

Skill-level, dimension-level, multi-dimensional, and coarse-grained evaluations of LLMs on FairytaleQA are presented in Table 1.

Skills within dimensions provide a good comparison point. Analyzing skill-level performance within a dimension, such as skill-level results in the scope dimension section of Table 1, allows for more meaningful comparisons between models or model variants, revealing trade-offs or uneven gains that overall metrics, such as coarse-grained results in the last column of Table 1, might mask. From the LLM-as-a-Judge-based results of Gemma-3 models on the FairytaleQA test set, we can observe an overall coarse-grained performance improvement from the 4B to the 12B model. However, disaggregating results by the Scope dimension reveals that this improvement is primarily driven by gains on local questions, while performance on global questions slightly declines. This finding suggests that increasing model size does not uniformly enhance all reasoning capabilities, showing the advantage of dimensional skill-level evaluation.

Dimension-level scores could indicate development priorities. The aggregated dimension-level scores (denoted as ***Dim*** in the Table 1) provide a high-level summary of model performance across different skill areas. These scores highlight which dimensions excel and which ones lag behind. Notably, the Element dimension consistently shows stronger performance compared to others, suggest-

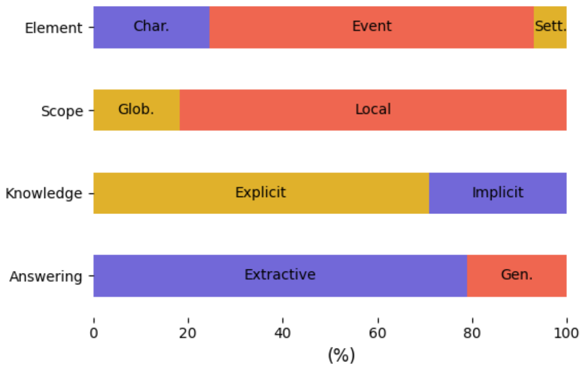


Figure 2: Skill imbalance in the test set of FairytaleQA

ing that models are better at recognizing common skill patterns within this dimension. While previous research (Bao et al., 2023; Peng et al., 2023) has primarily focused on enhancing the reasoning skills of language models by adding skill representations along a dimension similar to the Element dimension, our findings suggest that prioritizing the improvement of other underperforming dimensions could better address the models' developmental needs.

Multi-dimensional aggregates provide more realistic performance estimates. When aggregating performance across all dimensions (denoted as ***Multi-Dim*** in the Table), models achieve lower scores compared to evaluations based on a single coarse-grained metric (last row in the Table). For example, GPT-4o scores 91.90 based on the coarse-grained metric but only 89.72 under the multi-dimensional evaluation in the Table 1. This gap is even larger for smaller models like LLaMA-3 1B, which scores 59.46 in coarse-grained versus 55.77 in the multi-dimensional evaluation setting.

This discrepancy highlights that coarse-grained evaluations tend to overestimate model capabilities by masking weaknesses in specific skills. Moreover, skill imbalance across dimensions, such as in FairytaleQA shown in Figure 2, can further distort these coarse-grained metrics, as models may perform well on overrepresented or easier skills, inflating overall scores. The multi-dimensional aggregate, by capturing performance variability across underrepresented and challenging skill dimensions and skills, provides a more nuanced and accurate assessment of overall task performance. Consequently, multi-dimensional evaluation offers a more representative measure of model readiness for complex, real-world applications.

6 Related Works

Dimension-based evaluation has been explored in other NLP tasks such as summarization and open-ended dialogue, where metrics are typically defined by dimensions. For example, summarization is often assessed in terms of consistency, relevance, fluency, and coherence (Jain et al., 2023), while open-domain dialogue is evaluated along dimensions such as appropriateness, content, grammar, and relevance (Lin and Chen, 2023). In contrast, fine-grained evaluation of question answering has largely focused on selected skills (Sugawara et al., 2017a,b; Ye et al., 2024). Although these skills can be grouped into broader dimensions that reflect higher-level competencies, prior work has typically discussed such dimensions only at a conceptual level, without explicitly structuring evaluation or analysis around them (Schlegel et al., 2020; Rogers et al., 2023; Kalbaliyev and Sirts, 2024). Our work moves beyond selective skill-level assessment to demonstrate how organizing evaluation around dimensions yields clearer insights into model behavior and capabilities.

Another line of research focuses on analyzing the instance-level complexity of benchmarks, either to gain a deeper understanding of the skills being evaluated or to identify informative subsets of examples that better capture task diversity and LLM performance (Rodriguez et al., 2021; Ye et al., 2023; Gor et al., 2024; Cook et al., 2025). While our work emphasizes the importance of evaluating models across all core skills and dimensions, we leave for future research the exploration of how instance-level complexity analysis can be combined with dimensional skill evaluation.

7 Conclusion

This study argues that a dimension-focused skill evaluation offers a more accurate and insightful assessment of large language models on complex tasks like Narrative Question Answering. Unlike coarse-grained evaluation, this approach uncovers specific strengths and weaknesses across skill dimensions, revealing that model improvements are often uneven and concentrated in certain areas. The findings emphasize the limitations of coarse-grained evaluation and advocate for dimension-level analysis to guide model development priorities and better reflect real-world performance readiness.

Limitations

In this paper, we focus our analysis on a single skill taxonomy in order to clearly demonstrate the applicability of skill dimensions. While our evaluation method is inherently flexible and can be adapted to alternative taxonomies with similar structural properties, its reliance on the availability of a well-defined taxonomy poses a limitation. Applying the framework to new tasks may require the design or refinement of task-specific taxonomies, which can be both time-consuming and non-trivial. In particular, ensuring consistency, coverage, and granularity across different domains could introduce additional challenges.

Another limitation is related to the skill annotation. The process of labeling questions according to skill taxonomy inevitably depends on the expertise, perspective, and assumptions of the annotator. While the dimensional skill taxonomy is conceptually differentiable across multiple dimensions, the boundaries between individual skills within each dimension may be interpreted differently depending on the annotator. For example, determining whether a question requires implicit versus explicit reasoning often hinges on how much background knowledge the annotator, a large language model in our case, considers "commonsense." This introduces a degree of subjectivity and potential variability in the labeling process. Despite these limitations, the skill taxonomies provide a structured framework for categorizing questions and allow for systematic analysis of model performance across a range of skill types. Consequently, even with inherent labeling variability, it offers valuable insights into the strengths and weaknesses of models in handling diverse reasoning and knowledge-intensive tasks.

Additionally, since the primary objective of this work is to introduce and justify a dimensional skill evaluation methodology, we do not use computationally demanding, larger language models or reasoning models.

Acknowledgments

This work was supported by the Estonian Research Council Grant PSG721.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Meikai Bao, Qi Liu, Kai Zhang, Ye Liu, Linan Yue, Longfei Li, and Jun Zhou. 2023. Keep skills in mind: Understanding and implementing skills in commonsense question answering. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5012–5020. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2021. Evaluation of text generation: A survey. *Preprint*, arXiv:2006.14799.

Ryan A. Cook, John P. Lalor, and Ahmed Abbasi. 2025. No simple answer to data complexity: An examination of instance-level complexity metrics for classification tasks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2553–2573, Albuquerque, New Mexico. Association for Computational Linguistics.

FlowAI. 2024. Flow Judge: An Open Small Language Model for LLM System Evaluations.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

Maharshi Gor, Hal Daumé Iii, Tianyi Zhou, and Jordan Lee Boyd-Graber. 2024. Do great minds think alike? investigating human-AI complementarity in question answering with CAIMIRA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21533–21564, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8487–8495, Toronto, Canada. Association for Computational Linguistics.

Emil Kalbaliyev and Kairit Sirts. 2022. Narrative why-question answering: A review of challenges and datasets. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 520–530, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Emil Kalbaliyev and Kairit Sirts. 2024. On narrative question answering skills. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 814–820, Mexico City, Mexico. Association for Computational Linguistics.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 47–58, Toronto, Canada. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Alison H. Paris and Scott G. Paris. 2003. Assessing narrative comprehension in young children. *Reading Research Quarterly*, 38(1):36–76.

Wei Peng, Wanshui Li, and Yue Hu. 2023. Leader-generator net: Dividing skill and implicitness for conquering fairytaleqa. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 791–801, New York, NY, USA. Association for Computing Machinery.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Comput. Surv.*, 55(10).

Viktor Schlegel, Marco Valentino, Andre Freitas, Goran Nenadic, and Riza Batista-Navarro. 2020. A framework for evaluation of machine reading comprehension gold standards. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5359–5369, Marseille, France. European Language Resources Association.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017a. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 806–817, Vancouver, Canada. Association for Computational Linguistics.

Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. 2017b. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey

Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

University of Tartu. 2018. UT Rocket.

Leandro Von Werra, Lewis Tunstall, Abhishek Thakur, Sasha Luccioni, Tristan Thrush, Aleksandra Piktus, Felix Marty, Nazneen Rajani, Victor Mustar, and Helen Ngo. 2022. Evaluate & evaluation on the hub: Better best practices for data and model measurements. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 128–136, Abu Dhabi, UAE. Association for Computational Linguistics.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Qinyuan Ye, Harvey Fu, Xiang Ren, and Robin Jia. 2023. How predictable are large language model capabilities? a case study on BIG-bench. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7493–7517, Singapore. Association for Computational Linguistics.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *The Twelfth International Conference on Learning Representations*.

A Skills Annotation

As skills are categorized under distinct dimensions in the taxonomy proposed by Kalbaliyev and Sirts (2024), we classify questions separately based on each dimension. In the following subsections, we outline our chosen methods to facilitate efficient annotation.

A.1 Element Dimension Annotation

Xu et al. (2022) involves human annotators to label narrative elements from taxonomy of Paris and Paris (2003) for FairytaleQA dataset. Bao et al. (2023) use a set of common words or phrases, referred to as skill seeds, to automatically annotate questions in CommonsenseQA and CommonsenseQA 2.0 datasets (Talmor et al., 2019, 2021).

Since keyword-based annotation often fails to produce robust results, we opted to annotate the dataset along the element dimension by prompting a Large Language Model with element definitions. Specifically, we used GPT-4o (OpenAI et al., 2024) to classify the dataset. For this process, we provided the model with a System Prompt and passed both the context and the question as a User Prompt to guide the annotation.

System Prompt 1. Element Dimension

You are a helpful assistant. You will be provided with a context, and a question. Based on the definition below, determine if the question is Character, Event or Setting:

- It is **Event** question if it asks about an activity, an action, an event, or relationships between among events and characters, such as a reason. For example,'why', 'what happened?' questions are event questions.
- It is **Character** question if it directly asks about the identity, feeling or characteristics of the characters. For example, "who" is character question.
- It is **Setting** question if it asks about specific place, time, and environment in which the events take place. For example, "where" and "when" are setting questions.

For each question, provide a brief explanation of your reasoning and classify the question. Use the following format:
Explanation: <your explanation>
Classification: <Character or Event or Setting>

User Prompt 1. Annotation

CONTEXT: <context>
QUESTION: <question>

We generated responses via Azure OpenAI's API with the temperature value of 0. To validate our question annotations, we leveraged the existing annotated FairytaleQA dataset (Xu et al., 2022) as a reference point and approximately aligned its skill taxonomy with our chosen framework: Character (character and feeling), Event (causal relationship, outcome resolution, action), and Setting (setting). Using this alignment, the GPT-based annotation achieved an accuracy of 98% on the roughly mapped validation set from FairytaleQA.

A.2 Scope Dimension Annotation

To annotate questions along the scope dimension, it is important to first define what constitutes a story part. A story part can be interpreted as a sentence, a paragraph, a scene, or another unit of narrative segmentation. If sentences are used as the unit of segmentation, then single-sentence reasoning can be considered local, while multi-sentence reasoning can be attributed to global. If paragraphs or scenes are used as the unit of segmentation, especially in longer narratives, then local questions would be those that can be answered within a single paragraph or scene, while global questions would require information spanning across multiple paragraphs or scenes. The appropriate level of granularity depends largely on the nature of the dataset and the average length of the narratives involved. As we are focusing on short-form narratives, we chose local/global reasoning at the sentence level.

We prompted GPT-4o (OpenAI et al., 2024) to annotate the dataset based on the scope of text required to answer each question. We used the following System Prompt and passed context, question, and correct answers as a user prompt to annotate the dataset.

System Prompt 2. Scope Dimension

You are a helpful assistant. You will be provided with a context, a question, and the correct answer. Based on the definitions below, classify the question as either local or global:

- **Local** questions need information from only a single sentence to be answered correctly.
- **Global** questions need information from multiple sentences or the whole context to be answered correctly.

For each question, provide a brief explanation of your reasoning and classify the question. Use the following format:
Explanation: <your explanation>
Classification: <Local or Global>

We generated responses via Azure OpenAI's API with the temperature value of 0. To validate the classification accuracy, we used the skill-annotated portion of the Dream (Sun et al., 2019) dataset and mapped its skill taxonomy to our selected taxonomy. The GPT-based annotation achieved an accuracy and macro-F1 score of 91.5.

A.3 Knowledge Dimension Annotation

Similar to our approach for scope annotation, we prompted GPT-4o to classify datasets based on the implicitness of the question. We deliberately avoided providing a definition of explicitness, as doing so led to confusion between the two classes. To perform the annotation, we used the following System Prompt and supplied the context, question, and correct answer in the User Prompt.

System Prompt 3. Knowledge Dimension

You are a helpful assistant. You will be provided with a context, a question, and the correct answer. Based on the definition below, determine if the question is implicit:

- **Implicit** questions are those where the reader must use commonsense (world) knowledge or read between the lines to answer the question.

For each question, provide a brief explanation of your reasoning and classify the question. Use the following format:
Explanation: <your explanation>
Implicit: <Yes or No>

User Prompt 2. Annotation

CONTEXT: <context>
QUESTION: <question>
CORRECT ANSWERS: <answers>

To assess the classification accuracy, we evaluated the model on the skill-annotated portion of the DREAM dataset (Sun et al., 2019), aligning its original skill taxonomy with our selected taxonomy. The GPT-based annotation achieved an accuracy of 82.7% and a macro F1 score of 81%. Considering the inherent subjectivity in interpreting question implicitness, we found this result to be reasonable.

A.4 Answering Dimension Annotation

In order to annotate questions in the answering dimension, we automatically lemmatized and lowercased the context, question, and answers, removing articles and punctuation from the lemmas. Lemmatization was performed using spaCy's en_core_web_sm model. The context and question were then combined to form the input. If any lemmatized answer was a subset of the input lemmas, the question was classified as Extractive; otherwise, it was classified as Generative.

B Additional Details on Evaluation setup

B.1 Dataset

We selected the test set from the FairytaleQA (Xu et al., 2022) for evaluation. The questions were annotated according to the skill taxonomy proposed by Kalbaliyev and Sirts (2024), following the procedure detailed in Appendix A.

B.2 Language Model Prompting

We performed zero-shot prompting with Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, and Llama-3.1-8B-Instruct models (Grattafiori et al., 2024) and gemma-3-1b-it, gemma-3-4b-it, and gemma-3-12b-it models (Team et al., 2025) models, as well as the GPT-4o (OpenAI et al., 2024). For the Gemma-3 and Llama-3, we used the LM Evaluation Harness library (Gao et al., 2024) with models loaded from the Hugging Face Hub. We run inferences on Tesla V100 GPUs in the High Performance Computing Center of the University of Tartu (University of Tartu, 2018). For GPT-4o, we generated responses via Azure OpenAI's API. The following user prompt was used for question answering.

User Prompt 3. Question Answering Prompt

Answer the question based on the context. Keep your answer concise, few words are enough.
CONTEXT: <context>
QUESTION: <question>
ANSWER:

For all models, responses were generated using a temperature of 1.0 and a maximum new tokens value of 1024. The reported results are presented as the average of 5 model runs.

B.3 Metrics

We evaluated accuracy using an LLM-as-a-Judge to perform binary (Pass/Fail) assessments of generated outputs against reference answers. For this purpose, we used Flow Judge v0.1 (FlowAI, 2024), a specialized LLM-as-a-Judge model derived from further fine-tuning of the Phi-3.5-mini instruct model (Abdin et al., 2024). We selected Flow Judge v0.1 because of its compact size and performance comparable to the larger models, such as GPT-4o. This evaluation was conducted using the LM Evaluation Harness (Gao et al., 2024), with generation parameters used in the model's technical report: temperature of 0.1, Top P value of 0.95, and a maximum new tokens value of 1024. We used the following adapted scoring rubric:

- *Score 0 - Fail:* The generated response is completely incorrect or irrelevant to the query, with no overlap in information with any of the reference answers.
- *Score 1 - Pass:* The generated response matches one of the reference answers. The meaning conveyed by the generated response is equivalent to the reference. The generated response may leave out non-essential details compared to the references.

Additionally, we also evaluated the model with ROUGE-L (Lin, 2004) as an additional commonly used evaluation metric. We used the Hugging Face Evaluate library (Von Werra et al., 2022) implementation. The evaluation was conducted with the use_stemmer parameter set to True.

Language model performances based on LLM-as-a-Judge and ROUGE-L evaluation can be found in Tables 1 and 2.

C Results with ROUGE-L

Table 2 presents the ROUGE-L evaluation results. Although the scores differ from those obtained under the LLM-as-Judge setting, the main conclusions drawn from Table 1 remain valid. Analyzing performance at the skill level within each dimension, such as the element dimension section of Table 2, facilitates more fine-grained comparisons across models and model variants. This view makes it possible to identify trade-offs and uneven improvements that are often obscured when relying solely on aggregate metrics, such as those reported in the final column of Table 2.

Similarly, the element dimension consistently yields higher performance than other dimensions, suggesting that models are more adept at recognizing recurring skill patterns in this dimension. Additionally, with multi-dimensional evaluation, models achieve substantially lower scores compared to evaluations based only on a single coarse-grained metric. This discrepancy indicates that coarse-grained evaluations tend to overestimate model capabilities by concealing weaknesses in specific skills.

We note that the choice of metric for free-form evaluation can substantially influence the results. For instance, while LLM-as-a-judge-based evaluation of Gemma-3 models shows uneven gains in

	Models	Llama-3			Gemma-3			GPT-4o
Dimension	Skills/Size	1B	3B	8B	1b	4b	12b	-
Element	Character	36.76	55.27	62.68	54.62	60.47	65.58	66.86
	Event	36.06	49.36	55.27	37.66	42.53	44.96	60.09
	Setting	53.70	67.21	78.78	65.15	78.08	74.80	83.51
	Dim	***42.17***	***57.28***	***65.58***	***52.48***	***60.36***	***61.78***	***70.15***
Scope	Local	41.12	56.96	64.43	47.66	54.28	57.01	68.31
	Global	20.88	29.85	32.89	25.88	27.24	29.81	41.05
	Dim	***31.00***	***43.41***	***48.66***	***36.77***	***40.76***	***43.41***	***54.68***
Knowledge	Explicit	43.97	60.23	67.15	50.44	56.98	58.98	71.27
	Implicit	21.51	32.06	38.12	27.29	30.82	35.21	44.05
	Dim	***32.74***	***46.15***	***52.63***	***38.86***	***43.90***	***47.10***	***57.66***
Answering	Extractive	42.09	58.22	65.16	48.43	54.55	56.55	69.43
	Generative	19.90	28.71	34.34	25.85	29.81	35.16	40.43
	Dim	***31.00***	***43.47***	***49.75***	***37.14***	***42.18***	***45.85***	***54.93***
All	***Multi-Dim***	***34.23***	***47.58***	***54.16***	***41.31***	***46.80***	***49.54***	***59.36***
Coarse-grained		37.44	52.03	58.70	43.70	49.37	52.07	63.35

Table 2: Evaluation of Large Language Models on the FairytaleQA test set with the ROUGE-L metric. Dimension column indicates the sections of the table corresponding to specific skill dimensions. Skills show the skill-level evaluation of models. ***Dim*** refers to results of dimension-level evaluation, while ***Multi-Dim*** represents a multi-dimensional evaluation of the task. The last row provides a single coarse-grained evaluation.

the scope dimension, ROUGE-L scores indicate this difference in the element dimension. As LLM-as-a-judge offers greater flexibility in accommodating variations in model predictions, we report it as our main evaluation metric. Nevertheless, human evaluation remains the gold standard for text generation tasks, including Narrative QA (Celikyilmaz et al., 2021). However, conducting human evaluation is both costly and time-consuming (Lal et al., 2021), and its reliability has also been questioned (Gatt and Krahmer, 2018), particularly for QA instances that inherently involve multiple ambiguities (Kalbaliyev and Sirts, 2022).

D Data and Code Availability

The FairytaleQA dataset is publicly available on Hugging Face.[1] Code for annotation, inference, and evaluation is available on GitHub. [2]

---

[1] `https://huggingface.co/datasets/WorkInTheDark/FairytaleQA`

[2] `https://github.com/EmilKalbaliyev/Dimensional-Skill-Evaluation`

Potentially Problematic Word Usages and How to Detect Them: A Survey

Aina Garí Soler[1] **Matthieu Labeau**[2] **Chloé Clavel**[1]
[1]INRIA, Paris, [2]LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
`{aina.gari-soler,chloe.clavel}@inria.fr`, `matthieu.labeau@telecom-paris.fr`

Abstract

We introduce and frame the concept of potentially problematic word usages (PPWUs): word occurrences that are likely to cause communication breakdowns of a semantic nature. While much research has been devoted to lexical complexity, ambiguity, vagueness and related issues, no work has attempted to fully capture the intricate nature of PPWUs. We review linguistic factors, datasets and metrics that can be helpful for PPWU detection. We also discuss challenges to their study, such as their complexity and subjectivity, and highlight the need for future work on this phenomenon.

1 Introduction

Language is a powerful communication tool allowing us to exchange complex messages, but information is not always conveyed successfully. Miscommunication can be due to multiple factors, both linguistic and non-linguistic (e.g., environmental or psychological causes, such as a noisy background or the listener's lack of attention). In this paper, we focus on cases where a specific use of a word can give rise to a misunderstanding or an objection related to its meaning.

We define a **potentially problematic word usage (PPWU)** as an instance of a word[1] in context which is likely to cause some type of miscommunication (misunderstanding or non-understanding) or disagreement of a semantic nature.[2] For example, in "I enjoy working on my car", it is not clear whether *working* means polishing it as a hobby or repairing it (see Table 1). In practice, PPWUs should be determined with a specific target population in mind. In this paper, we discuss intrinsic factors of words and their contexts which are problematic both for specific communities but also for the general public (e.g., words with a false friend in a language may only risk being misunderstood or misused by speakers of that language, but a word in an underconstrained context can be problematic for anyone). We say, of a specific word usage, that it is an (actual) problematic word usage (PWU) when there is evidence that it has been misunderstood or disagreed upon by someone (e.g., when someone has signalled it in an interaction, asking "what do you mean by ...?" or similar questions (Noble et al., 2021)).

The detection of PPWUs has several applications. It can be useful for text simplification and readability assessment and can also have uses in applications related to language learning in general, such as aiding in choosing the right learning materials to adapt them to a student's level, or designing exercises that target specific types of PPWUs. If integrated with a conversational system, PPWUs could either be actively avoided in production, or trigger directed clarification requests to the user. Ensuring clear language use contributes to decrease misunderstandings, which can have negative psychological and physiological effects (Crockett et al., 2022). As part of a writing assistance tool, it could help identify words to be replaced to improve clarity. It could also be used to detect lexical errors in translated text.

Numerous studies focus on specific types or causes of PPWUs, but a unifying perspective encompassing them in their full complexity is missing. In this paper, we introduce and frame the notion of PPWU bringing together insights and points of view from research on different domains (e.g, psycholinguistics, computational linguistics, cognitive science, NLP) and with different goals, providing a foundation and framework for future research on PPWU detection. To this end, we compile and outline multiple reasons why a specific word or word usage is likely to cause miscommunication or be disagreed upon, thereby proposing a first character-

---

[1] We consider open-class lexemes or lexical items in general, but refer to them as "words" for simplicity.

[2] We exclude cases of unclear referents of referring expressions such as pronouns or proper nouns.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 441–463
November 8-9, 2025 ©2025 Association for Computational Linguistics

ization of PPWUs. We also present existing computational methods, resources and datasets that can be helpful for the study and detection of PPWUs, identifying areas where more work and knowledge is necessary; and include a discussion on various considerations to make when conducting research on PPWUs.

To compile the bibliography for this survey, we searched for relevant literature on Google Scholar using keywords such as "polysemy detection," "false friends," "lexical complexity," "text readability," or "lexical errors;" and expanded the search by examining papers citing, or cited by, the returned papers.

2 Factors contributing to PPWUs

In this section we provide a non-exhaustive[3] list of linguistic properties and phenomena linked to PPWUs. We distinguish between factors tied exclusively to word identity (Section 2.1) and factors linked to the linguistic context where a word is used (Section 2.2).[4] In the Appendix, we provide examples (Table 1) and a diagram summarizing the content of this section (Figure 1).

2.1 Context-independent factors

Some words are inherently difficult to understand, regardless of the context they appear in. This is often referred to as **lexical complexity** (North et al., 2023). Here, we outline word characteristics that may be *reasons* for a word not to be understood or for its meaning to be disagreed upon.[5] Note that the properties presented here are not absolute determinants of PPWUs; they often interact with each other as well as with other linguistic variables.[6] While we have aimed to define distinct categories where possible, they are not mutually exclusive and can co-occur in real-world usage.

2.1.1 Word properties

Lexical ambiguity. Words that have multiple senses (homonyms and polysemous words) are more likely to be misunderstood, even if the context is enough for disambiguation, because the audience might not be familiar with all of their senses, especially if some of them are not very frequent. The number of polysemous words and unique senses in a text has been found to correlate with its readability (Danilov et al., 2023), and the readers' knowledge of the multiple senses of a word correlates with reading comprehension (Kenneth Logan and Kieffer, 2017; Booton et al., 2022). Martínez Alonso et al. (2015) found that, in a sense annotation task, words with a higher number of senses and sense entropy tend to display higher disagreement. Words that have undergone lexical semantic changes (LSC) can be particularly difficult: readers or listeners are less likely to know novel senses of a word if these are recent, or, if reading a text from a different time period, a word may have changed in a way that the reader may not be aware of (e.g., *gay* used to mean *light-hearted, cheerful*). Approaches to quantifying a word's number of senses (**polysemy detection**, Garí Soler and Apidianaki (2021a)) and whether, how and when a word has changed meaning (**LSC detection**, Schlechtweg et al. (2020); Montariol et al. (2021)) can be helpful in finding words that are more likely to be misunderstood.

Word frequency is one of the most useful features to estimate lexical complexity (Specia et al., 2012; Wilkens et al., 2014; Garí et al., 2018). This is not surprising, as rare words are less likely to be part of a speaker's vocabulary, because they may have had less or no exposure to them. Acronyms, if not commonly used and not introduced properly in a text, are also a common source of confusion. Numerous studies have shown the relationship between word frequency and reading comprehension (Marks et al., 1974; Freebody and Anderson, 1983; Nouri and Zerhouni, 2018), confirming that rarer words are more prone to limit comprehension than more frequent words. Word frequency lists exist for multiple languages (Speer, 2022). Their source is very important: the nature, register, variety and size of the corpus, among other factors, may determine the usefulness of word frequency estimations as features for lexical complexity prediction (Wilkens et al., 2014).

---

[3] Our goal is to propose a categorization of PPWU causes linked to relevant NLP tasks, but there are other idiosyncratic causes of PPWUs that are difficult to categorize.

[4] This distinction bears some parallels to the notions of "meaning potential" and "situated meaning" (Myrendal, 2019).

[5] Some of the notions presented are used as features in lexical complexity detection. We exclude factors that, while correlated with complexity, are not likely to cause miscommunication on their own (e.g., word length and syllable count (Desai et al., 2021)).

[6] For example, a low-frequency word that is morphologically transparent (*cardiomyopathy*) may be easier to understand than another rare word with an opaque morphology (*gybe*).

Neologisms are newly created words recently introduced into a language, and which have a certain degree of acceptance in a linguistic community (e.g., *rizz*). As such, speakers (especially non-native ones (Charteris-Black, 1998)) are less likely to know them. An additional difficulty of neologisms is that they are often driven by technological advancements, so certain speakers may be unfamiliar not only with a new word form but also with the concept it expresses. At the same time, neologisms are often coined from existing morphemes (e.g., *mansplaining*) or from other languages, which may make their interpretation easier than that of neologisms created *ex nihilo* (e.g., *cromulent*) (Lehrer, 2003). Although less common, grammatical neologisms (existing words used with a new part of speech (PoS)) can also be problematic. Approaches to **neologism detection** (Janssen, 2012; Falk et al., 2014; Klosa and Lüngen, 2018) can be useful for detecting PPWUs. Finally, semantic neologisms (the appearance of new senses) are a form of lexical semantic change which can be addressed with the task of **novel sense detection** (Section 2.2).

Lexical variation refers to differences in lexical choice due to factors such as age, profession and social class (sociolects) or geographical location (dialectal variation). This includes slang, jargon and dialect-specific words which may only be produced and understood by certain communities of speakers. These differences can be particularly problematic in heterogeneous conversations (with people from, e.g., different age groups or cultures). Work on detecting **synchronic lexical differences** (Gonen et al., 2020; Yin et al., 2018; Schlechtweg et al., 2019; Garimella et al., 2016) and **dialect lexicon induction** (Scherrer, 2007; Artemova and Plank, 2023) could help identify words that are distinctive of certain groups of speakers.

Idioms are phrases with a non-compositional meaning; i.e., their meaning cannot be inferred from their parts (e.g., *kick the bucket*). As such, they need to be learned as autonomous lexical items. It is well known that idioms are particularly challenging for second language (L2) learners (Schraw et al., 1988; Alhaysony, 2017), except when a similar idiom exists in their native language (Irujo, 1986). Regardless of the frequency of an idiom, familiarity with the words that compose it can give a false perception of understanding. In fact, learners tend to overestimate their comprehension of idioms made up of high frequency words (Martinez and Murphy, 2011; Park and Chon, 2019); and when analyzing a sample of English idioms, Libben and Titone (2008) found that the frequency of verbs in idioms was negatively correlated with the predictability of their meaning (see Section 5 for more considerations on undetected misunderstanding). Relevant NLP tasks include **idiomatic expression identification** (Zeng and Bhat, 2021), the distinction between **literal and non-literal** idiom usages (Li and Sporleder, 2010), and **compositionality detection** (Cordeiro et al., 2019).

2.1.2 Concept-related properties

Complex meaning. Words designing complex concepts may be misunderstood because of a lack of or an incomplete knowledge of the reality being described (e.g., *enthalpy*). While estimating the complexity of a concept is a hard and subjective task, as it is not a well-established notion, **automatic term extraction** can be a good proxy to find words designating advanced or technical concepts (Hätty et al., 2020; Rigouts Terryn et al., 2020).

Vagueness and generality. Some words have an inherently vague meaning, i.e., a meaning that lacks precision (Van Rooij, 2011). This often concerns gradable adjectives, both relative (such as *big* or *tall*) and absolute (e.g., *bald*, *flat*), but it can also be found in words of other PoS (e.g., *heap*, *idiot*). These words describe qualities for which it is hard to draw a line and which can have multiple interpretations. Pezzelle and Fernández (2023) show that when faced when unclear gradable adjectives, speakers can increase their alignment with explicit interaction about word meaning. Adjectives may inherently be more problematic than other PoS because their semantic contribution to a noun is highly variable and combination-specific (Boleda et al., 2013).

Words that are very general, or high in a hypernymy hierarchy (e.g., *thing* or *do*), may not be specific enough and require clarification or more details. A mismatch between the provided level of specificity and the level required by the communicative situation, i.e., flouting Grice's maximum of quantity (Grice, 1975), can also generate confusion (Cruse, 1977).

Scalar adjective identification (Garí Soler and Apidianaki, 2021b) can be useful to identify potentially vague adjectives and distinguishing them from relational ones (e.g., *wooden*). The VAGO system (Icard et al., 2022) for measuring vagueness in texts relies on a database with lexical vagueness information (Atemezing et al., 2022). Lexicons and ontologies such as WordNet (Fellbaum, 1998) can serve as references for generality, and work on **semantic content quantification** (Herbelot and Ganesalingam, 2013; Santus et al., 2014) and more specifically on **hypernymy detection** (Shwartz et al., 2016; Cho et al., 2020) can help identify words with a general meaning.

Connotatively loaded terms. Words related to or referring to controversial topics are also prone to cause disagreement and misunderstanding: speakers may have different mental representations of these words, which are affected by their own opinions on the topic. For example, speakers may detect a disagreement on what constitutes or qualifies as *sexism* or *abuse* and signal this in a conversation (Myrendal, 2019). Work on **controversial topic detection** (Choi et al., 2010; Garimella et al., 2018) as well as on **lexico-semantic alignment** in, or outside of, conversations (Garí Soler et al., 2022, 2023) can potentially help detect this kind of words.

2.1.3 Cross-linguistic influence

In this section we describe cross-linguistic factors that may result in PPWUs when the situation involves L2 learners or bilingual individuals in a particular language pair.

False friends and partial cognates. Two words in two different languages are said to be false friends if they sound or look similar but have different meanings (e.g., *embarrassed* and Spanish *embarazada*, which means *pregnant*). A related concept is that of partial cognates, where the two similar words share some, but not all, meanings (see Table 1). These words can easily give rise to confusion if they have not been learned properly. Work on **false friend** or **partial cognate detection** (Inkpen et al., 2005; Mitkov et al., 2007; Ljubešić and Fišer, 2013; Palmero Aprosio et al., 2020; Lefever et al., 2020; Kanojia et al., 2021) can help identify words that may be problematic, both in terms of production (Raušer, 2017) and comprehension (Mattheoudakis and Patsala, 2007), when a speaker of a particular language is involved.

Cross-linguistic inequivalence. Every language offers different conceptualizations of the world and maps words to referents in different ways (Pavlenko, 2009). There is rarely a 100% translation equivalence between words in two languages, and often words that are almost equivalent differ in specific nuances that are hard to notice and to master, for language learners (Shalaby et al., 2009) but also bilingual speakers (Ameel et al., 2005). These differences can be quite notorious[7] or very subtle, such as what is reflected in the categorization and naming of similar objects (Malt et al., 2003; Pavlenko and Malt, 2011). This is sometimes referred to as "cross-linguistic near-synonymy" (Gries et al., 2020) and is studied in the fields of contrastive lexicology and lexico-semantic typology (Schapper and Koptjevskaja-Tamm, 2022). In other cases, for cultural or historical reasons, some terms in a language may not exist at all in another language, because they designate realities that do not exist in the other culture (e.g., Russian форточка (fortochka), a specific kind of small window for ventilation). These differences can result in unintended cross-linguistic transfer during production and comprehension by multilingual individuals (Jarvis, 2011), which can lead to misunderstandings. An NLP task that could assist in automatically finding cross-linguistic near-synonyms is **bilingual lexicon induction** (Irvine and Callison-Burch, 2017).

Words from another language. Sometimes multilingual speakers interject words from other languages which may be unknown to their interlocutors or readers. **Code switching detection** (Samih et al., 2016; Kevers, 2022) and **language identification** algorithms can detect these usages, particularly when they are tailored to the word level (Solorio et al., 2014; Rijhwani et al., 2017; Ansari et al., 2021).

2.2 Context-dependent factors

A word usage may also be misunderstood or disagreed upon because of the characteristics of the context in which it is used, even if the word itself is not usually problematic.[8] See Table 1 in the Appendix for examples.

Contextual underspecification. While many words have multiple senses, context is often enough for disambiguation, and in practice humans do not have a problem understanding many polysemous word usages (Piantadosi et al., 2012). When this

---

[7] *meat* is often translated as мясо (myaso) in Russian, but in its everyday use, мясо does not include poultry.

[8] It is, however, not possible to fully disentangle context from word identity: whether a word's context is problematic strongly depends on the word itself.

is not the case, however, ambiguity may require clarification. This happens with underconstrained contexts which do not provide enough information to establish the sense of a word, and which are a common reason of disagreement in word sense annotation, especially when allowing only the annotation of a single sense (Jurgens, 2014). There is not much work on identifying such ambiguities at the lexical level (Liu et al., 2023). Quantifying the **informativeness** of a context with respect to a specific target word could be helpful (Montariol et al., 2019; Schick and Schütze, 2019), but existing approaches assume that the target word is unknown.

Novel senses and metaphors. A speaker may use a word in a new sense that it has only recently acquired, or propose a creative or figurative use of an existing word. The latter are called novel metaphors (i.e., metaphors that do not rely on established conceptual mappings such as LOVE IS A JOURNEY (Lakoff, 1993)), and are harder to understand than conventional metaphors (Lai et al., 2009; Horvat et al., 2022). There is work on **unknown** or **novel word sense detection** (Erk, 2006; Cook et al., 2014), as well as **novel metaphor detection** (Schulder and Hovy, 2014; Haagsma and Bjerva, 2016; Reimann and Scheffler, 2024). One challenge is distinguishing these creative uses of language from errors, discussed below.

Lexical errors. An existing word may be inappropriately used instead of a correct alternative for multiple reasons. In text, a word can be confused with a homophone or another similar-sounding term, causing malapropisms (e.g., *insurance* for *assurance*). Sometimes a speaker may mix words that have a similar or related meaning (*broth* and *stock* or *trip* and *journey*) (Shalaby et al., 2009), or may use a different word because they can't come up with the correct one. Lexical errors can also be due to misspellings which result in another existing word (so-called "real-word errors," like *angel* for *angle*) (Azmi et al., 2019), or be caused by automatic correction tools. Of course, native and non-native speakers may make different mistakes. There is abundant work studying the types of lexical errors encountered in essays written by non-native speakers (Hemchua et al., 2006; Saud, 2018), but not so much about the kinds of mistakes made by native speakers other than malapropisms (Hirst et al., 1998), presumably because they are much rarer and harder to detect. Not all mistakes are equally confusing, however. While exposed to linguistic input, humans develop expectations about what is going to be said, and surface form similarity may facilitate understanding. There is also evidence that native speakers adapt their expectations when faced with non-native speech, and are more likely to find interpretations for implausible statements (Lev-Ari, 2015; Gibson et al., 2017).

Studies targeting anomalies at the lexical level aim at detecting text obfuscation (deliberate word substitutions to encrypt a message (Fong et al., 2008), for example to bypass censorship (Ji and Knight, 2018)); at identifying **real-word misspellings** (Samanta and Chaudhuri, 2013; Bravo-Candel et al., 2021) or detecting **miscollocations** (Wanner et al., 2013).

Rare senses. As discussed in Section 2.1.1, polysemous words used in a rare sense may be problematic. Precisely due to their low frequency, Word Sense Disambiguation (WSD) and Neural Machine Translation (NMT) systems also struggle with them (Campolungo et al., 2022a). Efforts toward **identifying rare word senses** (McCarthy et al., 2004) and correctly **disambiguating** them in context (Barba et al., 2021; Hangya et al., 2021; Campolungo et al., 2022b) can be helpful to find this kind of PPWUs.

Conversational Maxims Flouting. Going against conversational maxims (Grice, 1975) can cause confusion for the listener, especially the maxim of quality, with contradictory statements, jokes, information that goes against common sense, or **vandalism** in, for example, collaborative text editing (Adler et al., 2011). Approaches for **semantic plausibility** (Ko et al., 2019) and **joke detection** (Baranov et al., 2023) can be relevant for these usages, but they are typically designed to work at the sentence or text level.

3 Data

In this section we describe the kinds of data that can be used to investigate PPWUs or to train systems to automatically detect them. Our focus is on datasets of actual PWUs where there is evidence of speaker differences in word meaning. For datasets dedicated to the linguistic factors described in Section 2, refer to Table 2 in the Appendix. The most obvious clues come from real, spontaneous interactions where a speaker explicitly signals a problem with a word used in a conversation (Section 3.1).

Non-dialogical types of data exposing word usage or comprehension differences between speakers are presented in Sections 3.2 and 3.3.

3.1 Dialog

When miscommunication happens in dialog, if it is detected, it can be addressed by means of repair strategies, such as rephrasing or asking for clarification (Purver et al., 2003; Pickering and Garrod, 2004). While some datasets on clarification questions (Xu et al., 2019; Aliannejadi et al., 2019; Kumar and Black, 2020) and repair strategies exist (Rasenberg et al., 2022), they are rarely dedicated to problems with word meaning.

The most relevant work is on **word meaning negotiation** (WMN) in dialog, both in language learning settings (Varonis and Gass, 1985) as well as in online discussions (Myrendal, 2019). WMN can be understood as a case of conversational repair targeting the meaning of a specific word occurrence. WMN instances typically have three parts: a trigger (the PWU), an indicator (the turn signaling a clarification request or an objection), and the meaning negotiation, which is a metalinguistic discussion where speakers explicitly discuss the meaning of the trigger word (Varonis and Gass, 1985; Myrendal, 2015). Myrendal (2015) distinguishes two types of WMN: those which arise from an incomplete understanding and those caused by a disagreement about how someone has used a word. Currently, only one dataset with WMN annotations exists: the NeWMe corpus (Garí Soler et al., 2025a). NeWMe contains over 600 WMNs and related phenomena, coming from both oral and online conversations and involving a PWU. This kind of data can shed light on more aspects and characteristics of (P)PWUs. For example, in an analysis of triggers in NeWMe, Noble et al. (2025) found that disagreement problems tend to involve abstract terms, while understanding issues are more often linked to concrete terms. However, the data remains scarce; given the variety of PPWUs, more annotations covering a wider range of conversational situations are needed for their study.

3.2 Monolog

In monolog-like text, due to the absence of feedback, it is typically harder to anticipate what parts of the discourse may be unclear. In fact, speakers tend to overestimate their interlocutor's understanding (Keysar and Henly, 2002). Collaborative writing environments like Wikipedia, where texts are revised and edited by multiple authors, provide however a useful testbed to investigate problematic language use, by highlighting what can be improved in the original text. The wikiHowToImprove dataset (Anthonio et al., 2020) contains 2.7 million sentences with their revisions. The advantage of wikiHow compared to other sources of **revision histories** like Wikipedia (Faruqui et al., 2018) or news (Spangher et al., 2022) is that, in the former, modifications are more likely to be linguistically motivated instead of updating factual knowledge or providing additional information.

In this kind of interactive setting, PPWUs can be studied through word replacements. For example, Anthonio and Roth (2020) investigate noun substitutions in wikiHowToImprove. The fact that a word is replaced, however, is not evidence of it being a PWU: modifications are not always of a semantic nature (e.g., misspelling corrections), and only about 70% of revised versions in the dataset were judged to be improvements. Indeed, revisions may introduce vandalism, serve to just improve general textual coherence, or simply reflect the editor's personal lexical choice preferences (e.g., replacing *start* with *begin*). A dataset of word replacements annotated to indicate whether they constitute a semantic improvement is currently missing. Other kinds of modifications that can provide interesting data for the study of PPWUs are specific kinds of word insertions that clarify a word usage, such as completions of underspecified noun phrases (*tank* → *goldfish tank*) (Roth et al., 2022).

Finally, another useful kind of data could be obtained by directly asking annotators to mark, in a text, the word usages that they do not understand.

3.3 Signals from annotator disagreement

We can also find evidence of PWUs in cases of low inter-annotator agreement in semantic annotations at the word level. For example, in datasets where multiple people annotated senses (Jurgens and Klapaftis, 2013) or word usage similarity (Erk et al., 2009, 2013; Schlechtweg et al., 2021, 2025). Disagreement can be due to multiple reasons, such as inadequacy of the labels, but it can also point to the difficulty and subjectivity of the task (Plank, 2022), unveiling ambiguous, unclear or underspecified word usages. A tendency for a word to have lower levels of agreement can indicate a vaguer meaning (McCarthy et al., 2016) or other word characteristics that hinder its comprehension.

Low *intra*-annotator (Abercrombie et al., 2023) agreement is also worth exploring. Schober (2005) found evidence of "conceptual misalignment" (i.e., speakers having different mental representations of words) in surveys: Respondents taking the same survey twice were more likely to change their answers when provided with clarifications such as word definitions in the second iteration. This shows that there had been a misunderstanding the first time around. However, this kind of data is expensive to obtain and it requires formulating hypotheses about the words that are likely to be misunderstood.

4 Methods

We present here metrics, tools, features and methods that can be useful for PPWU prediction, for example because they have been shown to correlate with, or help predict, factors linked to PPWUs. We do not discuss supervised models tailored to the specific NLP tasks.

4.1 Language Model-derived measures

Language models (LMs) are trained with large amounts of text and can provide a measure of the likelihood of sentences or of words in context. Different LM-derived measures can be used to estimate the predictability of a word.

One commonly used measure is word **surprisal**, calculated as the negative log probability of a word occurrence. Surprisal theory (Hale, 2001) states that the cognitive effort required to process a word is proportional to its surprisal. This relationship has been largely studied, and surprisal has been found to be a generally good predictor of words' reading times (RT) (Smith and Levy, 2013; Goodkind and Bicknell, 2018); although it seems to better reflect RTs of lower verbal intelligence profiles (Haller et al., 2024). The choice of LM, however, is an important one. Surprisal values from large LMs with lower perplexities provide worse RT estimations than smaller models (Oh and Schuler, 2023).[9]

Another LM-derived measure is **entropy**. While surprisal is a measure of how unexpected a given word is in its context, entropy is calculated on the probability distribution over the vocabulary, elicited by the preceding context of a specific word position. It can be interpreted as the difficulty of predicting the continuation of a context. Pimentel et al. (2023) use entropy as an operationalization of anticipation and find that it can be a better predictor of RTs than surprisal.

One limitation of these measures for PPWU detection is that they are not only sensitive to (some) semantic anomalies but also to misspellings and syntactic deviations. One promising first step toward solving this problem, so far at the text level, is the contrastive perplexity score (Todd et al., 2020). The difference in perplexity between two LMs (one of which has been fine-tuned on a specific domain) gives information about the nature of the anomaly found (i.e., semantic vs non-semantic).

4.2 Word vector representations

Vector representations of words have been used for a long time in NLP. Static representations from Vector Space Models or word2vec (Mikolov et al., 2013), where a word is assigned a unique vector, have been shown to reflect multiple aspects of words' semantics. For instance, they can be used for identifying synchronic and diachronic lexical variation (Hamilton et al., 2016; Schlechtweg et al., 2019; Gonen et al., 2020), idioms (Peng and Feldman, 2017), hypernyms (Santus et al., 2014) and, in their multilingual form, false friends (Palmero Aprosio et al., 2020) (Section 2.1).

Contextualized word embeddings from Transformer-based language models like BERT (Devlin et al., 2019) additionally allow to represent word semantics at the token level and obtain good results on context-sensitive tasks (Section 2.2) such as novel metaphor detection (Pedinotti et al., 2021) and WSD (Wiedemann et al., 2019). The similarity between a target word and its context has been used to predict eye-tracking features (Salicchi et al., 2023). At the same time, these representations have been shown to encode rich out-of-context lexico-semantic information, such as a word's polysemy level, intensity, complexity and figurativeness (Garí Soler and Apidianaki, 2020; Xypolopoulos et al., 2021; Lyu et al., 2023).

4.3 Cognitive and neurolinguistic measures

Although costly to obtain, psycholinguistic and neurolinguisic data at the word level, as measured through **eye-tracking** and electroencefalography (EEG), can also help identify PPWUs. See Table 2 for datasets annotated with related metrics.

Longer eye fixations and reading times are typically interpreted as an indication of higher cognitive load (Just and Carpenter, 1980; Kintsch et al., 1975). Some eye tracking-derived measures, such

---

[9] The authors hypothesize that this is due to the vastly larger amount of training data compared to what humans are exposed to.

as gaze duration and first fixation duration, are affected by word frequency and predictability (Inhoff, 1984). Ambiguity also plays a role: Eye fixation times are longer on ambiguous words which have equally likely meanings (Rayner and Duffy, 1986) or for which previous context instantiated a less frequent meaning (Sereno et al., 1992).

The **N400** is an event-related potential signal in the brain consisting of a negative peak occurring about 400 milliseconds after a stimulus. It has been observed as a reaction to all sorts of semantically surprising input (Kutas and Hillyard, 1980; Kutas and Federmeier, 2011). Marked N400 waves have also been found in the processing of metaphorical language (Coulson, 2008). N400 has been shown to reflect word predictability modeled as LM surprisal (Michaelov et al., 2024), and evidence from joke processing shows that it is also sensitive to semantic plausibility operationalized as the similarity of a word with its preceding context (Xu et al., 2024). However, thematic role violations (e.g., between a verb and its argument: "Every morning at breakfast the eggs would eat...") seem to ellicit instead a P600 effect, typically linked with syntactic violations and ambiguities, but no N400 effect (Kuperberg, 2007).

4.4 Linguistic features

The earliest approaches to solving the NLP tasks presented in Section 2 often relied on different kinds and combinations of carefully selected linguistic features. For example, frequency measures (raw, comparative, or diachronic) can be useful for neologism detection (Garcia-Fernandez et al., 2011) and terminology extraction (Rigouts Terryn et al., 2020). PoS and syntactic features have been used to predict disagreement in sense annotations (Martínez Alonso et al., 2015). In lexical complexity prediction, multiple statistical, formal and psycholinguistic features (e.g., word length and concreteness) are also often used (Desai et al., 2021). Ngram information can be employed for neologism (Falk et al., 2014) and code-switching detection (Kevers, 2022), and word co-occurrence information for semantic content quantification (Herbelot and Ganesalingam, 2013). Finally, information on words' selectional preferences has been helpful for metaphor detection (Haagsma and Bjerva, 2016).

4.5 Other approaches

While it is not possible to present all existing methods here, there are other, less widespread approaches that are worth mentioning. For example, the **uncertainty of automatic sense annotations** from WSD models (Liu and Liu, 2023) can be used to find ambiguous word usages as well as to identify words with a tendency to present disambiguation difficulties. **Anomaly** or **outlier detection** methods, although they are most often applied to texts and not at the word level (Ruff et al., 2019; Arora et al., 2021), have been used to detect novel as well as figurative and metaphoric word usages (Sasaki and Shinnou, 2012; Bejan et al., 2023) and idioms (Feldman and Peng, 2013). **Topic modeling** has been used for novel word sense detection (Lau et al., 2014) and idiom detection (Peng et al., 2014); and **sentiment analysis** can be useful for controversial topic detection (Choi et al., 2010). There has also been work evaluating the ability of generative Large LMs to detect ambiguities, not restricted to lexical ones (Liu et al., 2023); but LLMs' effectiveness to detect different kinds of PPWUs remains unexplored.

5 Open Directions and Challenges

As we have shown, there is a substantial amount of work on detecting individual phenomena or word characteristics associated with PPWUs. For the study of communicative success and failure, as well as for the outlined applications of PPWU detection, we argue that it is worth aspiring to address PPWUs in their full diversity, encompassing the multiple factors presented, but no such approach currently exists.[10] The present survey aims to provide an initial overarching perspective that sets the ground for future work.

One key reason for the lack of a comprehensive approach is the **complexity and the varied nature** of PPWUs. The most effective solution may involve a combination of multiple approaches tailored to specific purposes, audiences, situations, or PPWU types. A big gap preventing progress is the **scarcity of large, good-quality datasets** annotated with real PWUs. Annotation of WMNs is helpful but costly; recent efforts striving toward their semi-automatic annotation (Garí Soler et al., 2025b) will contribute to a better understanding of the characteristics of PWUs and their rate of

---

[10] Lexical complexity prediction is closely related, but it has a narrower scope. It typically focuses on concept-related difficulties that can cause non-understanding. Our definition of PPWUs is broader and more context-dependent, including usages that cause confusion and disagreement, and considers phenomena such as humor, ambiguity, and lexical errors.

occurrence, and will provide more data that could eventually serve to train models for (P)PWU detection. Existing metrics and methodology (Section 4) typically address a particular phenomenon, but it is not always clear what kinds of PPWUs they may detect, which ones they may fail to capture, and what kind of false positives they may propose. For example, LM-derived measures can find unexpected usages of words, which is practical for, e.g., lexical error detection, but would probably not be useful to identify contextual underspecification problems.

Another difficulty lies in the **subjectivity of PPWUs**. As mentioned in Section 1, what constitutes an actual PWU depends on various characteristics of both the speaker and the listener, such as their age, language proficiency, and any potential language impairments or pathologies. It is therefore important to take annotator characteristics into account when collecting data for PPWU detection, and more data and studies are needed to determine the types of usages that are problematic for each type of audience. Differences are not necessarily restricted to specific groups – individual variation can occur even within a community, such as among language learners (Degraeuwe and Goethals, 2024). While the individual component should not be neglected, and it may be desirable to restrict the scope to usages problematic for a specific population, it is also possible to identify usages that are problematic across communities. Such usages may involve errors, underspecification, or jokes; where the issue comes from the production side and depends less on the listener's characteristics.

Unsignaled PPWUs present an additional challenge. Not all PPWUs result in communication breakdowns: Many misunderstandings and disagreements are not signaled, either because they go undetected or because they do not pose a real problem to continue communication. A consequence of this is that when collecting evidence of PPWUs in conversation, an undetermined number of PPWUs are prone to be overlooked. Identifying **keywords** within a dialog may help determine the words that are critical for the conversation to move forward. In a conversational system, this could help restrict the words for which clarification is needed. It is possible that PPWUs worthy of being signaled are more likely to be located in the parts of utterances that introduce new information (i.e., the comment, rather than the topic or theme, in information structure (Lambrecht, 1994)).

Although the perspective proposed here is intended to be language-independent, there may also be **language- and culture-specific** causes of PPWUs, and ways of adapting metrics and tools to specific languages. Whether a PPWU is signaled in a conversation may have a cultural component and be influenced by different politeness norms, potentially causing differences in the frequency with which they are observed across languages.

6 Conclusion

We have introduced the notion of Potentially Problematic Word Usages (PPWUs) and reviewed work from various disciplines to provide a comprehensive perspective of related linguistic factors, methods for their detection, and the kinds of data that can be used for their study. We have also discussed challenges to take into account when working with PPWUs, and identified areas for future work.

PPWUs are a complex and understudied phenomenon. They are affected by multiple factors, both linguistic (word and contextual properties) and non-linguistic (e.g., interlocutor characteristics and differences between them). More work is needed to broaden our understanding of PPWUs, their underlying causes and their prevalence, but datasets containing evidence of miscommunication due to specific word usages are scarce.

One of our long-term goals is to develop methods aimed at capturing PPWUs in their many forms, rather than focusing on just one subtype. With this work, we have laid out a framework organizing existing knowledge and methods, pointing out the diversity of problems that must be considered when tackling PPWUs, in order to serve as a stepping stone for future research on this multifaceted phenomenon.

Limitations

While we have made an effort to provide a comprehensive and cross-disciplinary overview of phenomena, datasets and approaches relevant to the study of problematic word usages, some relevant works may have been missed. Even regardless of the methodology for bibliography search, however, this survey cannot be fully exhaustive. For instance, we have identified and presented discrete factors associated with PPWUs, but word usages can be problematic due to other idiosyncratic reasons that may be hard to classify and detect (see last row of Table 1 for an example). Moreover, it remains unclear how often each of the presented factors actually leads to a PWU.

In this paper, partly due to space constraints, we have chosen to give only a high-level overview of the different research directions that need to be explored for studying PPWUs. Our primary goal was to establish a conceptual and empirical foundation for the study of PPWUs, which required an extensive synthesis of prior work. While we point to several avenues for future research, taking the first concrete steps, such as carrying out corpus-based studies that could propose a complete typology of PPWUs, is left for future work.

Finally, we acknowledge that, if taken naively, the definition of PPWUs is very broad and many words in a text could be considered to be "potentially problematic." However, this is precisely why we argue that PPWU identification must be audience- and context-sensitive. Bringing these factors together under a single perspective is crucial, because in real-life interaction, problems may arise from different sources, and effectively determining the underlying cause requires considering them jointly.

Acknowledgments

We thank all anonymous reviewers for their feedback. This research was partially funded by the Agence Nationale de la Recherche SINNet project (ANR-23-CE23-0033-01). Additional support was provided by the ANR under the France 2030 program PRAIRIE (ANR-23-IACL-0008).

References

Gavin Abercrombie, Verena Rieser, and Dirk Hovy. 2023. Consistency is key: Disentangling label variation in natural language processing with intra-annotator agreement. *arXiv preprint arXiv:2301.10684*.

B Thomas Adler, Luca De Alfaro, Santiago M Mola-Velasco, Paolo Rosso, and Andrew G West. 2011. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In *Computational Linguistics and Intelligent Text Processing: 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part II 12*, pages 277–288. Springer.

Maha H Alhaysony. 2017. Strategies and Difficulties of Understanding English idioms: A Case Study of Saudi University EFL Students. *International Journal of English Linguistics*, 7(3):70–84.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Eef Ameel, Gert Storms, Barbara C Malt, and Steven A Sloman. 2005. How bilinguals solve the naming problem. *Journal of memory and language*, 53(1):60–80.

Mohd Zeeshan Ansari, MM Sufyan Beg, Tanvir Ahmad, Mohd Jazib Khan, and Ghazali Wasim. 2021. Language Identification of Hindi-English tweets using code-mixed BERT. In *2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 248–252. IEEE.

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A Resource and Analyses on Edits in Instructional Texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Talita Anthonio and Michael Roth. 2020. What Can We Learn from Noun Substitutions in Revision Histories? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1359–1370, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Udit Arora, William Huang, and He He. 2021. Types of Out-of-Distribution Texts and How to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ekaterina Artemova and Barbara Plank. 2023. Low-resource Bilingual Dialect Lexicon Induction with Large Language Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 371–385, Tórshavn, Faroe Islands. University of Tartu Library.

Ghislain Atemezing, Benjamin Icard, and Paul Egré. 2022. Vague Terms in SKOS to detect vagueness in textual documents (Version v2).

Aqil M Azmi, Manal N Almutery, and Hatim A Aboalsamh. 2019. Real-word errors in Arabic texts: A better algorithm for detection and correction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1308–1320.

Alexander Baranov, Vladimir Kniazhevsky, and Pavel Braslavski. 2023. You Told Me That Joke Twice: A Systematic Investigation of Transferability and Robustness of Humor Detection Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13701–13715, Singapore. Association for Computational Linguistics.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with Extractive Sense Comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. DIACR-Ita @ EVALITA2020: Overview of the evalita2020 diachronic lexical semantics (diacr-ita) task. *Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*.

Matei Bejan, Andrei Manolache, and Marius Popescu. 2023. AD-NLP: A Benchmark for Anomaly Detection in Natural Language Processing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10766–10778, Singapore. Association for Computational Linguistics.

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 35–46, Potsdam, Germany. Association for Computational Linguistics.

Sophie A Booton, Alex Hodgkiss, Sandra Mathers, and Victoria A Murphy. 2022. Measuring knowledge of multiple word meanings in children with english as a first and an additional language and the relationship to reading comprehension. *Journal of Child Language*, 49(1):164–196.

Daniel Bravo-Candel, Jésica López-Hernández, José Antonio García-Díaz, Fernando Molina-Molina, and Francisco García-Sánchez. 2021. Automatic Correction of Real-Word Errors in Spanish Clinical Texts. *Sensors*, 21(9):2893.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli. 2022a. DiBiMT: A Novel Benchmark for Measuring Word Sense Disambiguation Biases in Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352, Dublin, Ireland. Association for Computational Linguistics.

Niccolò Campolungo, Tommaso Pasini, Denis Emelin, and Roberto Navigli. 2022b. Reducing Disambiguation Biases in NMT by Leveraging Explicit Word Sense Information. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4824–4838, Seattle, United States. Association for Computational Linguistics.

Jonathan Charteris-Black. 1998. Compound Nouns and the Acquisition of English Neologisms. *ERIC Document Reproduction service No. ED427535*.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023. ChiWUG: A Graph-based Evaluation Dataset for Chinese Lexical Semantic Change Detection. In *Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change*, pages 93–99, Singapore. Association for Computational Linguistics.

Yejin Cho, Juan Diego Rodriguez, Yifan Gao, and Katrin Erk. 2020. Leveraging WordNet Paths for Neural Hypernym Prediction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3007–3018, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng. 2010. Identifying Controversial Issues and their Subtopics in News Articles. In *Intelligence and Security Informatics: Pacific Asia Workshop, PAISI 2010, Hyderabad, India, June 21, 2010. Proceedings*, pages 140–153. Springer.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The VNC-Tokens Dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 19–22. Citeseer.

Paul Cook, Jey Han Lau, Diana McCarthy, and Timothy Baldwin. 2014. Novel Word-sense Identification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1624–1635, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch. 2019. Unsupervised Compositionality Prediction of Nominal Compounds. *Computational Linguistics*, 45(1):1–57.

Seana Coulson. 2008. Metaphor comprehension and the brain. *The Cambridge handbook of metaphor and thought*, pages 177–194.

Erin E Crockett, Monique MH Pollmann, and Ana P Olvera. 2022. You just don't get it: The impact of misunderstanding on psychological and physiological health. *Journal of Social and Personal Relationships*, 39(9):2847–2868.

D Alan Cruse. 1977. The pragmatics of lexical specificity. *Journal of linguistics*, 13(2):153–164.

Andrew V Danilov, Zilya Nuretdinova, Zulfat Miftakhutdinov, Elvira Sharifullina, Nazym Kydyrbayeva, and Tat'Yana Soldatkina. 2023. Polysemy as a Complexity Predictor in school textbooks. In *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, pages 721–725. IEEE.

Jasper Degraeuwe and Patrick Goethals. 2024. LexComSpaL2: A Lexical Complexity Corpus for Spanish as a Foreign Language. In *Proceedings of the*

2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 10432–10447, Torino, Italia. ELRA and ICCL.

Abhinandan Tejalkumar Desai, Kai North, Marcos Zampieri, and Christopher Homan. 2021. LCP-RIT at SemEval-2021 Task 1: Exploring Linguistic Features for Lexical Complexity Prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 548–553, Online. Association for Computational Linguistics.

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

Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. 2018. Weeding out Conventionalized Metaphors: A Corpus of Novel Metaphor Annotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1424, Brussels, Belgium. Association for Computational Linguistics.

Pedro J Chamizo Domínguez and Brigitte Nerlich. 2002. False friends: their origin and semantics in some selected languages. *Journal of pragmatics*, 34(12):1833–1849.

Katrin Erk. 2006. Unknown word sense detection as outlier detection. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 128–135, New York City, USA. Association for Computational Linguistics.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on Word Senses and Word Usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring Word Meaning in Context. *Computational Linguistics*, 39(3):511–554.

Ingrid Falk, Delphine Bernhard, and Christophe Gérard. 2014. From Non Word to New Word: Automatically Identifying Neologisms in French Newspapers. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4337–4344, Reykjavik, Iceland. European Language Resources Association (ELRA).

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Anna Feldman and Jing Peng. 2013. Automatic detection of idiomatic clauses. In *International conference on intelligent text processing and computational linguistics*, pages 435–446. Springer.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

SzeWang Fong, Dmitri Roussinov, and David B Skillicorn. 2008. Detecting Word Substitutions in Text. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1067–1076.

Peter Freebody and Richard C Anderson. 1983. Effects on text comprehension of differing proportions and locations of difficult vocabulary. *Journal of Reading Behavior*, 15(3):19–39.

Richard Futrell, Edward Gibson, Harry J Tily, Idan Blank, Anastasia Vishnevetsky, Steven T Piantadosi, and Evelina Fedorenko. 2021. The Natural Stories corpus: a reading-time corpus of English texts containing rare syntactic constructions. *Language Resources and Evaluation*, 55:63–77.

Anne Garcia-Fernandez, Anne-Laure Ligozat, Marco Dinarelli, and Delphine Bernhard. 2011. When was it Written? Automatically Determining Publication Dates. In *String Processing and Information Retrieval: 18th International Symposium, SPIRE 2011, Pisa, Italy, October 17-21, 2011. Proceedings 18*, pages 221–236. Springer.

Aina Garí, Marianna Apidianaki, and Alexandre Allauzen. 2018. A comparative study of word embeddings and other features for lexical complexity detection in French. In *Actes de la Conférence TALN. Volume 1 - Articles longs, articles courts de TALN*, pages 499–508, Rennes, France. ATALA.

Aina Garí Soler and Marianna Apidianaki. 2020. BERT Knows Punta Cana is not just beautiful, it's gorgeous: Ranking Scalar Adjectives with Contextualised Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7371–7385, Online. Association for Computational Linguistics.

Aina Garí Soler and Marianna Apidianaki. 2021a. Let's Play Mono-Poly: BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

Aina Garí Soler and Marianna Apidianaki. 2021b. Scalar Adjective Identification and Multilingual Ranking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4653–4660, Online. Association for Computational Linguistics.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2022. One Word, Two Sides: Traces of Stance in Contextualized Word Representations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3950–3959, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2023. Measuring Lexico-Semantic Alignment in Debates with Contextualized Word Representations. In *Proceedings of the First Workshop on Social Influence in Conversations (SICon 2023)*, pages 50–63, Toronto, Canada. Association for Computational Linguistics.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2025b. Toward the Automatic Detection of Word Meaning Negotiation Indicators in Conversation. *Accepted at Findings of the Association for Computational Linguistics: EMNLP 2025*.

Aina Garí Soler, Jenny Myrendal, Chloé Clavel, and Staffan Larsson. 2025a. The NeWMe Corpus: A gold standard corpus for the study of Word Meaning Negotiation. *PREPRINT (Version 1) available at Research Square*.

Aparna Garimella, Rada Mihalcea, and James Pennebaker. 2016. Identifying Cross-Cultural Differences in Word Usage. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 674–683, Osaka, Japan. The COLING 2016 Organizing Committee.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Quantifying Controversy on Social Media. *ACM Transactions on Social Computing*, 1(1):1–27.

Edward Gibson, Caitlin Tan, Richard Futrell, Kyle Mahowald, Lars Konieczny, Barbara Hemforth, and Evelina Fedorenko. 2017. Don't Underestimate the Benefits of Being Misunderstood. *Psychological science*, 28(6):703–712.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, Interpretable and Stable Method for Detecting Words with Usage Change across Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 538–555, Online. Association for Computational Linguistics.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Stefan Th Gries, Marlies Jansegers, and Viola G Miglio. 2020. Quantitative methods for corpus-based contrastive linguistics. In *New approaches to contrastive linguistics: empirical and mathodological challenges*, volume 336, pages 53–84. de Gruyter.

Hessel Haagsma and Johannes Bjerva. 2016. Detecting novel metaphor using selectional preference information. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 10–17, San Diego, California. Association for Computational Linguistics.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A Large Corpus of Potentially Idiomatic Expressions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 279–287, Marseille, France. European Language Resources Association.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Patrick Haller, Lena Bolliger, and Lena Jäger. 2024. Language models emulate certain cognitive profiles: An investigation of how predictability measures interact with individual differences. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7878–7892, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Viktor Hangya, Qianchu Liu, Dario Stojanovski, Alexander Fraser, and Anna Korhonen. 2021. Improving Machine Translation of Rare and Unseen Word Senses. In *Proceedings of the Sixth Conference on Machine Translation*, pages 614–624, Online. Association for Computational Linguistics.

Anna Hätty, Dominik Schlechtweg, Michael Dorna, and Sabine Schulte im Walde. 2020. Predicting Degrees of Technicality in Automatic Terminology Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2883–2889, Online. Association for Computational Linguistics.

Saengchan Hemchua, Norbert Schmitt, et al. 2006. An analysis of lexical errors in the English compositions of Thai learners. *PROSPECT-ADELAIDE-*, 21(3):3.

Aurélie Herbelot and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Sofia, Bulgaria. Association for Computational Linguistics.

Graeme Hirst, David St-Onge, et al. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a Lexical Simplifier Using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland. Association for Computational Linguistics.

Ana Werkmann Horvat, Marianna Bolognesi, Jeannette Littlemore, and John Barnden. 2022. Comprehension of different types of novel metaphors in monolinguals and multilinguals. *Language and Cognition*, 14(3):401–436.

Benjamin Icard, Ghislain Atemezing, and Paul Égré. 2022. VAGO: un outil en ligne de mesure du vague et de la subjectivité. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle (PFIA 2022)*, pages 68–71.

Albrecht Werner Inhoff. 1984. Two stages of word processing during eye fixations in the reading of prose. *Journal of verbal learning and verbal behavior*, 23(5):612–624.

Diana Inkpen, Oana Frunza, and Grzegorz Kondrak. 2005. Automatic Identification of Cognates and False Friends in French and English. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, volume 9, pages 251–257.

Suzanne Irujo. 1986. Don't put your Leg in your Mouth: Transfer in the Acquisition of Idioms in a Second Language. *tesol Quarterly*, 20(2):287–304.

Ann Irvine and Chris Callison-Burch. 2017. A Comprehensive Analysis of Bilingual Lexicon Induction. *Computational Linguistics*, 43(2):273–310.

Maarten Janssen. 2012. NeoTag: a POS Tagger for Grammatical Neologism Detection. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2118–2124, Istanbul, Turkey. European Language Resources Association (ELRA).

Scott Jarvis. 2011. Conceptual transfer: Crosslinguistic effects in categorization and construal. *Bilingualism: Language and cognition*, 14(1):1–8.

Heng Ji and Kevin Knight. 2018. Creative Language Encoding under Censorship. In *Proceedings of the First Workshop on Natural Language Processing for Internet Freedom*, pages 23–33, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3006–3012, Reykjavik, Iceland. European Language Resources Association (ELRA).

David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299, Atlanta, Georgia, USA. Association for Computational Linguistics.

Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87(4):329.

Diptesh Kanojia, Prashant Sharma, Sayali Ghodekar, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2021. Cognition-aware Cognate Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292, Online. Association for Computational Linguistics.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

J Kenneth Logan and Michael J Kieffer. 2017. Evaluating the role of polysemous word knowledge in reading comprehension among bilingual adolescents. *Reading and Writing*, 30:1687–1704.

Laurent Kevers. 2022. CoSwID, a Code Switching Identification Method Suitable for Under-Resourced Languages. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 112–121, Marseille, France. European Language Resources Association.

Boaz Keysar and Anne S Henly. 2002. Speakers' overestimation of their effectiveness. *Psychological Science*, 13(3):207–212.

Walter Kintsch, Ely Kozminsky, William J Streby, Gail McKoon, and Janice M Keenan. 1975. Comprehension and recall of text as a function of content variables. *Journal of verbal learning and verbal behavior*, 14(2):196–214.

Annette Klosa and Harald Lüngen. 2018. New German words: Detection and description. In *Proceedings of the XVIII EURALEX International Congress Lexicography in Global Contexts*, pages 559–569, Ljubljana. Znanstvena založba Filozofske fakultete Univerze v Ljubljani/Ljubljana University Press, Faculty of Arts.

Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2019. Linguistically-Informed Specificity and Semantic Plausibility for Dialogue Generation. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3456–3466, Minneapolis, Minnesota. Association for Computational Linguistics.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. SemEval-2013 Task 5: Evaluating Phrasal Semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 39–47, Atlanta, Georgia, USA. Association for Computational Linguistics.

Vaibhav Kumar and Alan W Black. 2020. ClarQ: A large-scale and diverse dataset for Clarification Question Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7296–7301, Online. Association for Computational Linguistics.

Gina R Kuperberg. 2007. Neural mechanisms of language comprehension: Challenges to syntax. *Brain research*, 1146:23–49.

Marta Kutas and Kara D Federmeier. 2011. Thirty years and counting: finding meaning in the N400 component of the event-related brain potential (ERP). *Annual review of psychology*, 62(1):621–647.

Marta Kutas and Steven A Hillyard. 1980. Event-related brain potentials to semantically inappropriate and surprisingly large words. *Biological psychology*, 11(2):99–116.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. NorDiaChange: Diachronic Semantic Change Dataset for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Vicky Tzuyin Lai, Tim Curran, and Lise Menn. 2009. Comprehending conventional and novel metaphors: An ERP study. *Brain research*, 1284:145–155.

George Lakoff. 1993. *The contemporary theory of metaphor*, page 202–251. Cambridge University Press.

Knud Lambrecht. 1994. *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents*, volume 71. Cambridge university press.

Jey Han Lau, Paul Cook, Diana McCarthy, Spandana Gella, and Timothy Baldwin. 2014. Learning Word Sense Distributions, Detecting Unattested Senses and Identifying Novel Senses Using Topic Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 259–270, Baltimore, Maryland. Association for Computational Linguistics.

John Lee and Chak Yan Yeung. 2018. Personalizing Lexical Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Els Lefever, Sofie Labat, and Pranaydeep Singh. 2020. Identifying Cognates in English-Dutch and French-Dutch by means of Orthographic Information and Cross-lingual Word Embeddings. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4096–4101, Marseille, France. European Language Resources Association.

Adrienne Lehrer. 2003. Understanding trendy neologisms. *Italian Journal of Linguistics*, 15:369–382.

Shiri Lev-Ari. 2015. Comprehending non-native speakers: Theory and evidence for adjustment in manner of processing. *Frontiers in psychology*, 5:1546.

Linlin Li and Caroline Sporleder. 2010. Linguistic Cues for Distinguishing Literal and Non-Literal Usages. In *Coling 2010: Posters*, pages 683–691, Beijing, China. Coling 2010 Organizing Committee.

Maya R Libben and Debra A Titone. 2008. The multi-determined nature of idiom processing. *Memory & cognition*, 36:1103–1121.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're Afraid Language Models Aren't Modeling Ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Zhu Liu and Ying Liu. 2023. Ambiguity Meets Uncertainty: Investigating Uncertainty Estimation for Word Sense Disambiguation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3963–3977, Toronto, Canada. Association for Computational Linguistics.

Nikola Ljubešić and Darja Fišer. 2013. Identifying false friends between closely related languages. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 69–77, Sofia, Bulgaria. Association for Computational Linguistics.

Steven G Luke and Kiel Christianson. 2018. The Provo Corpus: A large eye-tracking corpus with predictability norms. *Behavior research methods*, 50:826–833.

Qing Lyu, Marianna Apidianaki, and Chris Callison-burch. 2023. Representation of Lexical Stylistic Features in Language Models' Embedding Space. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 370–387, Toronto, Canada. Association for Computational Linguistics.

Mounica Maddela and Wei Xu. 2018. A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760, Brussels, Belgium. Association for Computational Linguistics.

Barbara C Malt, Steven A Sloman, and Silvia P Gennari. 2003. Universality and language specificity in object naming. *Journal of memory and language*, 49(1):20–42.

Carolyn B Marks, Marleen J Doctorow, and Merlin C Wittrock. 1974. Word Frequency and Reading Comprehension. *The Journal of Educational Research*, 67(6):259–262.

Ron Martinez and Victoria A Murphy. 2011. Effect of Frequency and Idiomaticity on Second Language Reading Comprehension. *Tesol Quarterly*, 45(2):267–290.

Héctor Martínez Alonso, Anders Johannsen, Oier Lopez de Lacalle, and Eneko Agirre. 2015. Predicting word sense annotation agreement. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 89–94, Lisbon, Portugal. Association for Computational Linguistics.

Marina Mattheoudakis and Paschalia Patsala. 2007. English-Greek false friends: Now they are, now they aren't. In *8th International Conference of Greek Linguistics. University of Ioannina, August 30th-September 7th*.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. 2016. Word Sense Clustering and Clusterability. *Computational Linguistics*, 42(2):245–275.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Automatic Identification of Infrequent Word Senses. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1220–1226, Geneva, Switzerland. COLING.

James A Michaelov, Megan D Bardolph, Cyma K Van Petten, Benjamin K Bergen, and Seana Coulson. 2024. Strong Prediction: Language model surprisal explains multiple N400 effects. *Neurobiology of language*, 5(1):107–135.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint:1301.3781v3*.

Ruslan Mitkov, Viktor Pekar, Dimitar Blagoev, and Andrea Mulloni. 2007. Methods for extracting and classifying pairs of cognates and false friends. *Machine translation*, 21:29–53.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Syrielle Montariol, Aina Garí Soler, and Alexandre Allauzen. 2019. Exploring sentence informativeness. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume II : Articles courts*, pages 303–312, Toulouse, France. ATALA.

Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and Interpretable Semantic Change Detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

Jenny Myrendal. 2015. *Word meaning negotiation in online discussion forum communication*. Ph.D. thesis, University of Gothenburg.

Jenny Myrendal. 2019. Negotiating meanings online: Disagreements about word meaning in discussion forum communication. *Discourse studies*, 21(3):317–339.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Daiki Nishihara and Tomoyuki Kajiwara. 2020. Word Complexity Estimation for Japanese Lexical Simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3114–3120, Marseille, France. European Language Resources Association.

Bill Noble, Staffan Larsson, and Jenny Myrendal. 2025. Misunderstanding the Concrete, Disagreeing About the Abstract: A Closer Look at Word Meaning Negotiation Triggers. In *Proceedings of the 29th Workshop on the Semantics and Pragmatics of Dialogue – Full Papers*, pages 81–91, Bielefeld, Germany. SEMDIAL.

Bill Noble, Kate Viloria, Staffan Larsson, and Asad Sayeed. 2021. What do you mean by negotiation? Annotating social media discussions about word meaning. In *Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2021)*.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

Nadia Nouri and Badia Zerhouni. 2018. Lexical Frequency Effect on Reading Comprehension and Recall. *Arab World English Journal (AWEJ) Volume*, 9.

Byung-Doh Oh and William Schuler. 2023. Why Does Surprisal From Larger Transformer-Based Language Models Provide a Poorer Fit to Human Reading Times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Jenny A Ortiz-Zambranoa and Arturo Montejo-Ráezb. 2020. Overview of ALexS 2020: First Workshop on Lexical Analysis at SEPLN. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*, volume 2664, pages 1–6.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Alessio Palmero Aprosio, Stefano Menini, and Sara Tonelli. 2020. Adaptive Complex Word Identification through False Friend Detection. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 192–200.

Jinkyoung Park and Yuah V Chon. 2019. EFL Learners' Knowledge of High-frequency Words in the Comprehension of Idioms: A Boost or a Burden? *RELC journal*, 50(2):219–234.

Aneta Pavlenko. 2009. *The bilingual mental lexicon: Interdisciplinary approaches*, volume 70. Multilingual Matters.

Aneta Pavlenko and Barbara C Malt. 2011. Kitchen Russian: Cross-linguistic differences and first-language object naming by Russian–English bilinguals. *Bilingualism: Language and Cognition*, 14(1):19–45.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The Language Demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2(Feb):79–92.

Paolo Pedinotti, Eliana Di Palma, Ludovica Cerini, and Alessandro Lenci. 2021. A howling success or a working sea? Testing what BERT knows about metaphors. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 192–204, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jing Peng and Anna Feldman. 2017. Automatic idiom recognition with word embeddings. In *Information Management and Big Data: Second Annual International Symposium, SIMBig 2015, Cusco, Peru, September 2-4, 2015, and Third Annual International Symposium, SIMBig 2016, Cusco, Peru, September 1-3, 2016, Revised Selected Papers 2*, pages 17–29. Springer.

Jing Peng, Anna Feldman, and Ekaterina Vylomova. 2014. Classifying Idiomatic and Literal Expressions Using Topic Models and Intensity of Emotions. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2019–2027, Doha, Qatar. Association for Computational Linguistics.

Sandro Pezzelle and Raquel Fernández. 2023. Semantic Adaptation to the Interpretation of Gradable Adjectives via Active Linguistic Interaction. *Cognitive Science*, 47(2):e13248.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Martin J Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, 27(2):169–190.

Tiago Pimentel, Clara Meister, Ethan G. Wilcox, Roger P. Levy, and Ryan Cotterell. 2023. On the Effect of Anticipation on Reading Times. *Transactions of the Association for Computational Linguistics*, 11:1624–1642.

Barbara Plank. 2022. The "Problem" of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2003. On the Means for Clarification in Dialogue. In R. Smith and J. van Kuppevelt, editors, *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*, pages 235–255. Kluwer Academic Publishers.

Marlou Rasenberg, Wim Pouw, Asli Özyürek, and Mark Dingemanse. 2022. The multimodal nature of communicative efficiency in social interaction. *Scientific Reports*, 12(1):19111.

Daniel Raušer. 2017. Selected English-Czech False Friends and Their Use in the Works of Some Czech Students. *Caracteres*, Vol. 6.

Keith Rayner and Susan A Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & cognition*, 14(3):191–201.

Sebastian Reimann and Tatjana Scheffler. 2024. When is a Metaphor Actually Novel? Annotating Metaphor Novelty in the Context of Automatic Metaphor Detection. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 87–97, St. Julians, Malta. Association for Computational Linguistics.

Ayla Rigouts Terryn, Veronique Hoste, Patrick Drouin, and Els Lefever. 2020. TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 85–94, Marseille, France. European Language Resources Association.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating Code-Switching on Twitter with a Novel Generalized Word-Level Language Detection Technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1037–1047, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael Roth, Talita Anthonio, and Anna Sauer. 2022. SemEval-2022 Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1039–1049, Seattle, United States. Association for Computational Linguistics.

Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft. 2019. Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071, Florence, Italy. Association for Computational Linguistics.

Lavinia Salicchi, Emmanuele Chersoni, and Alessandro Lenci. 2023. A study on surprisal and semantic relatedness for eye-tracking data prediction. *Frontiers in Psychology*, 14:1112365.

Pratip Samanta and Bidyut B. Chaudhuri. 2013. A simple real-word error detection and correction using local word bigram and trigram. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*, pages 211–220, Kaohsiung, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas. Association for Computational Linguistics.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing Hypernyms in Vector Spaces with Entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Minoru Sasaki and Hiroyuki Shinnou. 2012. Detection of Peculiar Word Sense by Distance Metric Learning with Labeled Examples. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 601–604, Istanbul, Turkey. European Language Resources Association (ELRA).

Wafa Ismail Saud. 2018. Lexical Errors of Third Year Undergraduate Students. *English Language Teaching*, 11(11):161–168.

Antoinette Schapper and Maria Koptjevskaja-Tamm. 2022. Introduction to special issue on areal typology of lexico-semantics. *Linguistic Typology*, 26(2):199–209.

Yves Scherrer. 2007. Adaptive String Distance Measures for Bilingual Dialect Lexicon Induction. In *Proceedings of the ACL 2007 Student Research Workshop*, pages 55–60, Prague, Czech Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2019. Attentive Mimicking: Better Word Embeddings by Attending to Informative Contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 489–494, Minneapolis, Minnesota. Association for Computational Linguistics.

Dominik Schlechtweg, Tejaswi Choppa, Wei Zhao, and Michael Roth. 2025. CoMeDi Shared Task: Median Judgment Classification & Mean Disagreement Ranking with Ordinal Word-in-Context Judgments. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 33–47, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A Framework for the Annotation of Lexical Semantic Change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael F Schober. 2005. Conceptual Alignment in Conversation. *Other minds: How humans bridge the divide between self and others*, pages 239–252.

Gregory Schraw, Woodrow Trathen, Ralph E Reynolds, and Richard T Lapan. 1988. Preferences for Idioms: Restrictions due to Lexicalization and Familiarity. *Journal of Psycholinguistic Research*, 17:413–424.

Marc Schulder and Eduard Hovy. 2014. Metaphor Detection through Term Relevance. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 18–26, Baltimore, MD. Association for Computational Linguistics.

Sara C Sereno, Jeremy M Pacht, and Keith Rayner. 1992. The effect of meaning frequency on processing lexically ambiguous words: Evidence from eye fixations. *Psychological Science*, 3(5):296–301.

Nadia A Shalaby, Noorchaya Yahya, and Mohamed El-Komi. 2009. Analysis of Lexical Errors in Saudi College Students' Compositions. *Journal of the Saudi Association of Languages and Translation*, 2(3):65–93.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — A New Corpus for Lexical Complexity Prediction from Likert Scale Data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. SemEval-2021 Task 1: Lexical Complexity Prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the First Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. 2022. NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 127–157, Seattle, United States. Association for Computational Linguistics.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 Task 1: English Lexical Simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. Association for Computational Linguistics.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Caroline Sporleder, Linlin Li, Philip Gorinski, and Xaver Koch. 2010. Idioms in Context: The IDIX Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*, volume 14. Amsterdam: John Benjamins.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning Arguments: Interaction Dynamics and Persuasion Strategies in Good-faith Online Discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624.

Graham Todd, Catalin Voss, and Jenny Hong. 2020. Unsupervised Anomaly Detection in Parole Hearings using Language Models. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 66–71, Online. Association for Computational Linguistics.

Kathryn K Toffolo, Edward G Freedman, and John J Foxe. 2022. Evoking the N400 Event-related Potential (ERP) Component Using a Publicly Available Novel Set of Sentences with Semantically Incongruent or Congruent Eggplants (Endings). *Neuroscience*, 501:143–158.

Robert Van Rooij. 2011. Vagueness and Linguistics. In *Vagueness: A guide*, pages 123–170. Springer.

Evangeline Marlos Varonis and Susan Gass. 1985. Non-native/non-native conversations: A model for negotiation of meaning. *Applied linguistics*, 6(1):71–90.

Leo Wanner, Serge Verlinde, and Margarita Alonso Ramos. 2013. Writing assistants and automatic lexical error correction: word combinatorics. *Electronic lexicography in the 21st century: Thinking outside the paper. Proceedings of eLex 2013*, pages 472–487.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Rodrigo Wilkens, Alessandro Dalla Vecchia, Marcely Zanon Boito, Muntsa Padró, and Aline Villavicencio. 2014. Size does not matter. Frequency does. A study of features for measuring lexical complexity. In *Advances in Artificial Intelligence–IBERAMIA 2014: 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings 14*, pages 129–140. Springer.

Haoyin Xu, Masaki Nakanishi, and Seana Coulson. 2024. Revisiting Joke Comprehension with Surprisal and Contextual Similarity: Implication from N400 and P600 Components. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking Clarification Questions in Knowledge-Based Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629, Hong Kong, China. Association for Computational Linguistics.

Christos Xypolopoulos, Antoine Tixier, and Michalis Vazirgiannis. 2021. Unsupervised Word Polysemy Quantification with Multiresolution Grids of Contextual Embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3391–3401, Online. Association for Computational Linguistics.

Yusuf Yaylaci and Arman Argynbayev. 2014. English-Russian False Friends in ELT Classes with Intercultural Communicative Perspectives. *Procedia-Social and Behavioral Sciences*, 122:58–64.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A Report on the Complex Word Identification Shared Task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Zi Yin, Vin Sachidananda, and Balaji Prabhakar. 2018. The Global Anchor Method for Quantifying Linguistic Shifts and Domain Adaptation. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic Expression Identification using Semantic Compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.

A Appendix

B PPWU contributing factors: Diagram

Cause	Example	Source
Low frequency	*Yes, I am a **byword** to them.*	Shardlow et al. (2020)
Neologism	*– I, yeah, I-m, yeah, I do aerobics, uh, **step classes** and, uh* *– Step classes?* *– toning classes, yes.* (At the time of speaking, 1992, stepping classes were a new concept)	Godfrey et al. (1992); Garí Soler et al. (2025a)
Lexical variation	*chips* denotes different things in UK and American English, and *lorry* is mainly exclusively used in the UK.	
Complex meaning	*It is unlikely that morphological changes during development led to impairment of spatial learning and motor coordination, and morphological alterations in the cytoarchitecture of the **hippocampus** and cerebellum were not observed* (...)	Shardlow et al. (2020)
Vagueness	*– Do you genuinely believe that the less palatable parts of TRP* [The Red Pill] *are less **effective** than the positive parts?* *– You'll have to define "effective" for me and give me a specific example or two* (...)*. Many forms of abuse, manipulation, and deception might be described as "effective", depending on what your goal is and whether or not you're a psychopath.*	Tan et al. (2016); Garí Soler et al. (2025a)
Generality	*– I wonder what kind of **care** you get when you're hospitalised for appendicitis, gallstone, ileus, inflammation of the pancreas or inflammation of the intestines.* (...) *– What do you mean by care? I was brought medicine and received help with mixing the nutritional drink which was replacing food during the time my intestines were resting. When I started feeling better and could move around I had to fix that myself.*	Myrendal (2015)
Connotatively loaded term	– (...) ***agnosticism** postulates that the existence or nonexistence of god is beyond our knowledge or ability to gather it.* (...) *– Not necessarily, or in most cases. Agnosticism argues that it is not possible at a given moment in time to know absolutely, but then we don't know anything absolutely. Moreover, that that's ok, we'll work with what we have.* (...) – (...) *That's kind of a flimsy sort of agnosticism* (...)	Tan et al. (2016); Garí Soler et al. (2025a)
False friend	*We usually go to a **magazine** to buy milk.* (Russian магазин (magazin) or French *magasin* mean *shop, store*)	Yaylaci and Argynbayev (2014)
Partial cognate	English *blank* and its equivalent in several Romance languages (ES: *blanco*, FR: blanc, PT: branco, IT: bianco, CA: blanc). They can both mean *empty* but the Romance versions also mean *white*.	Domínguez and Nerlich (2002)
Cross-linguistic inequivalence	Russian форточка (fortochka), a specific kind of small window for ventilation common in post-Soviet states.	Pavlenko (2009)
Word from another language	– (...) *the truth is that I don't know what the **gringo** fandom is like* (...) – (...) *Not sure what you mean by gringo* (...) *– Oh JAJFJWJF SORRY- "gringo" is a way that people who speak spanish refer to people who speak english* (...)	Noble et al. (2021)
Ambiguity or contextual underspecification	*Rooms are classically decorated and **warm***	Jurgens (2014)
Ambiguity or contextual underspecification	– (...) *I enjoy **working** on my car* (...) (...) *– Oh I thought you meant "working on my car" as in polishing it and keeping it in super condition as a hobby, not as in "occasional repairs".*	Tan et al. (2016); Garí Soler et al. (2025a)
Novel sense	The use of *bet* to mean *yes*	
Metaphor	*Westerns have a **gladiatorial**, timeless quality.*	Do Dinh et al. (2018)
Lexical error	*Choose which **charter** you want to be .* (correction: character)	Anthonio et al. (2020)
Rare sense	*– Anyone who has a link to a **dirty** Win7 download?* *– What do you mean by dirty?* *– What I meant by 'dirty' was an illegal copy of the operating system.*	Myrendal (2015)
Vandalism	*First , make sure your hamster is familiar with your scent and your **poop** .* (original word: "voice")	Anthonio and Roth (2020)
Semantic plausibility / contradiction	*i understand. i am not sure if i can afford a babysitter, i am a **millionaire***	Ko et al. (2019)
Unclassified	– (...) *True **waffles** are crisp on the outside and fluffy on the inside* (...) *– So the definition of waffle changes based on how long you cook it? I happen to enjoy my waffles slightly undercooked, does that mean that they are not waffles?*	Tan et al. (2016); Garí Soler et al. (2025a)

Table 1: Examples of PPWUs.

Task/Annotation	**Reference**	**Language**	**Resource name (if existing)**
Lexical Semantic Change	Schlechtweg et al. (2018)	DE	DURel
	Schlechtweg et al. (2020)	EN, DE, SV, LA	SemEval 2020 Task 1
	Basile et al. (2020)	IT	DIACR-Ita
	Rodina and Kutuzov (2020)	RU	RuSemShift
	Kutuzov et al. (2022)	NO	NorDiaChange
	Chen et al. (2023)	ZH	ChiWUG
Complex Word Identification	Horn et al. (2014)	EN	
	Paetzold and Specia (2016)	EN	CWI–2016
	Yimam et al. (2018)	EN, DE, ES	CWI–2018
	Maddela and Xu (2018)	EN	Word Complexity Lexicon
	Lee and Yeung (2018)	EN	
	Ortiz-Zambranoa and Montejo-Ráezb (2020)	ES	ALexS
	Nishihara and Kajiwara (2020)	JA	
	Shardlow et al. (2021)	EN	CompLex
Idiomatic Expressions	Haagsma et al. (2020)	EN	MAGPIE
	Korkontzelos et al. (2013)	EN	SemEval-2013 Task 5
	Cook et al. (2008)	EN	VNC-Tokens
	Sporleder et al. (2010)	EN	IDIX
Term Extraction	Rigouts Terryn et al. (2020)	EN, FR, NL	ACTER
Scalar Adjective Identification	Garí Soler and Apidianaki (2021b)	EN	SCAL-REL
Vague terms	Atemezing et al. (2022)	EN, FR	
False friends and cognates	Palmero Aprosio et al. (2020)	IT-{EN,FR,DE,ES}	
Bilingual Lexicon Induction	Pavlick et al. (2014)	100 languages	
Code-switching	Solorio et al. (2014)	MSA-DA, EN-ES, EN-ZH, EN-NE	
	Molina et al. (2016)	MSA-DA, EN-ES	
	Kevers (2022)	CO-FR	BDLC
	Nayak and Joshi (2022)	HI-EN	L3Cube-HingCorpus
Novel Word Sense Detection	Cook et al. (2014)	EN	BNC-ukWaC & SiBol/Port
Metaphor Detection	Steen et al. (2010)	EN	VUAMC
Novel Metaphor Detection	Do Dinh et al. (2018)	EN	
Self-paced reading times	Smith and Levy (2013)	EN	Brown Corpus
Self-paced reading times	Futrell et al. (2021)	EN	Natural Stories Corpus
Word-level eye-tracking	Kennedy et al. (2003)	EN	Dundee Corpus
Word-level eye-tracking	Luke and Christianson (2018)	EN	Provo Corpus
N400	Toffolo et al. (2022)	EN	

Table 2: Selection of datasets annotated with linguistic information relevant for PPWUs.

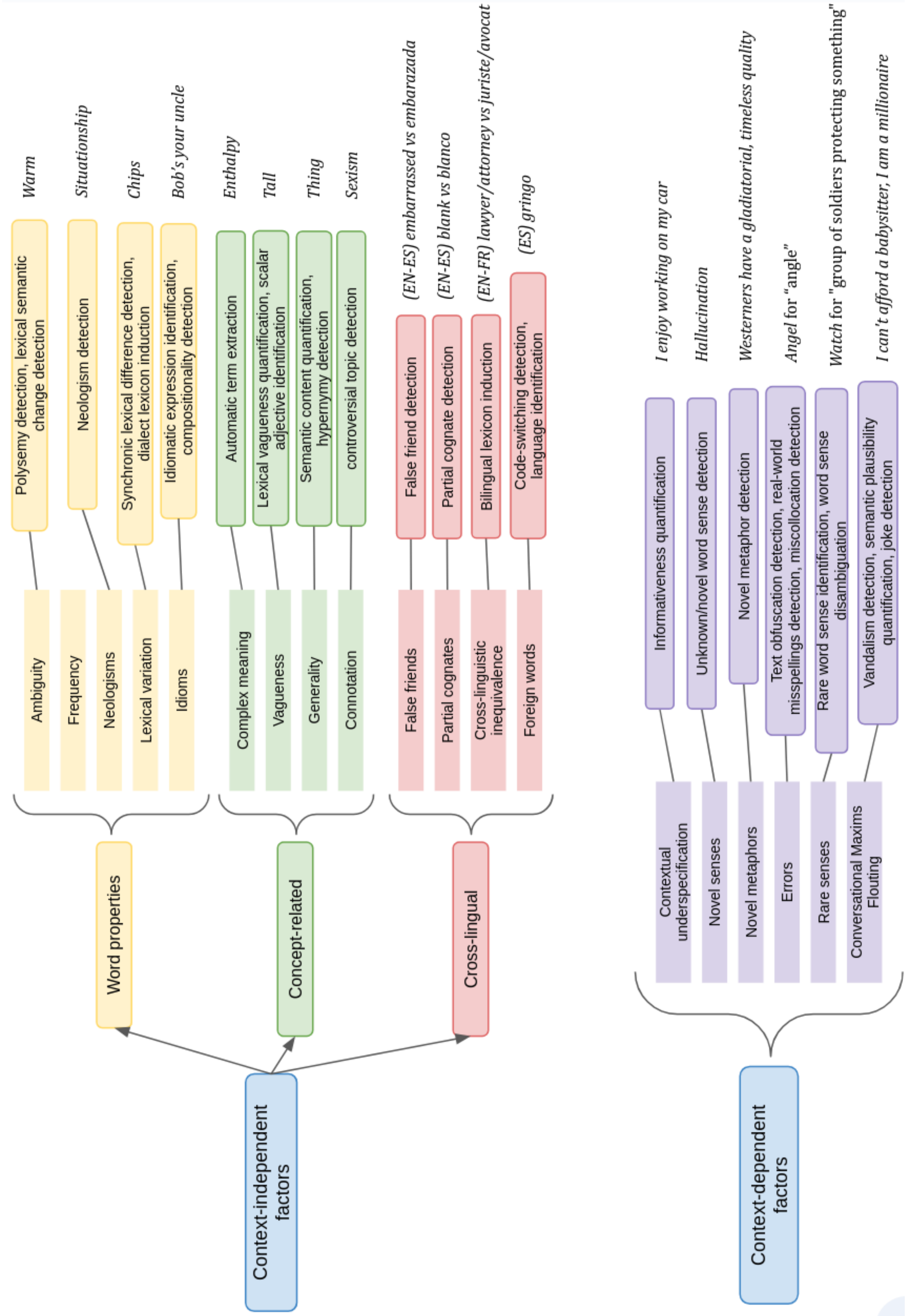

Context-independent factors
Word properties
Concept-related
Cross-lingual
Ambiguity
Frequency
Neologisms
Lexical variation
Idioms
Complex meaning
Vagueness
Generality
Connotation
False friends
Partial cognates
Cross-linguistic inequivalence
Foreign words
Polysemy detection, lexical semantic change detection
Neologism detection
Synchronic lexical difference detection, dialect lexicon induction
Idiomatic expression identification, compositionality detection
Automatic term extraction
Lexical vagueness quantification, scalar adjective identification
Semantic content quantification, hypernymy detection
controversial topic detection
False friend detection
Partial cognate detection
Bilingual lexicon induction
Code-switching detection, language identification
Warm
Situationship
Chips
Bob's your uncle
Enthalpy
Tall
Thing
Sexism
(EN-ES) embarrassed vs embarazada
(EN-ES) blank vs blanco
(EN-FR) lawyer/attorney vs juriste/avocat
(ES) gringo
Context-dependent factors
Contextual underspecification
Novel senses
Novel metaphors
Errors
Rare senses
Conversational Maxims Flouting
Informativeness quantification
Unknown/novel word sense detection
Novel metaphor detection
Text obfuscation detection, real-world misspellings detection, miscollocation detection
Rare word sense identification, word sense disambiguation
Vandalism detection, semantic plausibility quantification, joke detection
I enjoy working on my car
Hallucination
Westerners have a gladiatorial, timeless quality
Angel for "angle"
Watch for "group of soldiers protecting something"
I can't afford a babysitter, I am a millionaire

Figure 1: Factors associated with PPWUs and related NLP tasks, with examples.

Author Index

Abbaszadeh, Mina, 316
Abuín, Marta Vázquez, 172
Agichtein, Eugene, 395
Alavi, Seyed Hossein, 13
Arase, Yuki, 381

Bailleux, Thomas, 409
Basili, Roberto, 31
Bi, Yaxin, 48
Bolic, Miodrag, 70
Boulbarss, Bilal, 122
Bouraoui, Zied, 409

Cao, Zhihan, 134
Castro-Garcia, Juan Luis, 267
Ceron, Tanise, 83
Chandradevan, Ramraj, 395
Chang, Haw-Shiuan, 225
Chersoni, Emmanuele, 1
Clavel, Chloé, 441
Coeur, Christophe, 348
Constant, Mathieu, 348
Croce, Danilo, 31

Daffara, Agnese, 83
Dattawad, Sourabh, 83
Delor, Mariona Taulé, 59
Deshmukh, Advait, 189
Dessi, Roberto, 212
Dhole, Kaustubh, 395
Dumitru, Razvan-Gabriel, 280

Farrús, Mireia, 59
Fengkai, Liu, 293
Ferraro, Francis, 253, 304
Francoeur, Vincent, 70
Frassinelli, Diego, 142

Gagnon, Sylvain, 70
Gao, Qiyue, 179
Garcia, Marcos, 172
Garí Soler, Aina, 441
Gasan, Carol-Luca, 362
Gehring, Janosch, 152
Gorodissky, Yuval, 100
Guo, Rongchen, 70

Han, Tonghui, 48

Hasegawa, Ryo, 338
Hawashin, Hala, 316
Hu, Hai, 179

Kadam, Maithili Sanjay, 304
Kalbaliyev, Emil, 430
Kamigaito, Hidetaka, 338
Kang, Taekyu, 13
Kelious, Abdelhak, 348
Kordjamshidi, Parisa, 267
Kovalev, Pavel, 179
Krishna, Shambhavi, 225

Labeau, Matthieu, 441
Lee, John Sie Yuen, 293
Lee, Taesung, 225
Lewis, Martha, 122
Limbäck-Stokin, Tilen Gaetano, 316
Lindahl, Anna, 242
Liu, Dong, 328
Liu, Yikang, 179
Liu, Zhe, 13
Lo, Kin Ian, 316

Ma, Xinyue, 59
Magnini, Bernardo, 212
Marquis, Pierre, 409
Min, Junghyun, 202
Moss, Lawrence S., 179
Mukherjee, Tanmoy, 409
Mulvenna, Maurice D., 48

Naber, Sven, 142
Nejadgholi, Isar, 70
Nissim, Malvina, 212

Ogasa, Yuya, 381

Pacheco, Maria Leonor, 189, 372
Padó, Sebastian, 83
Pais, Vasile, 362
Paiva, Valeria de, 179
Pastells, Pol, 59
Pearson, Beth, 122

Qian, Zhiheng, 179
Qiu, Le, 1

Rahimi, Mahdi, 280
Rai, Shahid Iqbal, 31
Roth, Dan, 100
Roth, Michael, 152
Roy Dipta, Shubhashis, 253

Sadrzadeh, Mehrnoosh, 316
Sakai, Yusuke, 338
Schulte Im Walde, Sabine, 142
Shwartz, Vered, 13
Sirts, Kairit, 430
Srinivas, Dananjay, 189
Sudharsan, Bagyasree, 372
Sulem, Elior, 100
Surdeanu, Mihai, 280

Tokunaga, Takenobu, 134

Umadi, Ashwin, 189

Villavicencio, Aline, 1

Wang, Haoyu, 13
Watanabe, Taro, 338
Wazni, Hadi, 316
Wein, Shira, 202
Wray, Michael, 122

Yamada, Hiroaki, 134
Yang, Dongqiang, 48
Yang, Xiulin, 202
Yu, Yanxuan, 328

Zaninello, Andrea, 212
Zhang, Xinru, 48